



TUGAS AKHIR - KI091391

**RANCANG BANGUN PERMAINAN WAKTU
NYATA “YUK KITA CARI” DENGAN BANYAK
PEMAIN MENGGUNAKAN FLASH
ACTIONSCRIPT 3 (AS3) DAN
SMARTFOXSERVER2X (SFS2X)**

Mohammad F K Umam
NRP 5106100 064

Dosen Pembimbing
Imam Kuswardayan S.Kom., M.T.
Sarwosri S.Kom., M.T.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI091391

DESIGN AND IMPLEMENTATION OF REAL-TIME MULTIPLAYER GAME “YUK KITA CARI” USING FLASH ACTIONSCRIPT 3 (AS3) AND SMARTFOXSERVER2X (SFS2X)

Mohammad F K Umam
NRP 5106 100 064

Advisor
Imam Kuswardayan S.Kom., M.T.
Sarwosri S.Kom., M.T.

INFORMATICS ENGINEERING DEPARTEMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2014

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**Rancang Bangun Permainan Waktu Nyata “Yuk Kita Cari”
dengan Banyak Pemain Menggunakan Flash ActionScript 3
(AS3) dan SmartFoxServer2X (SFS2X)**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

MOHAMMAD F K UMAM

NRP: 5106 100 064

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Imam Kuswardayan S.Kom

NIP: 197612152003121

Pembimbing 1)

Sarwosri S.Kom., M.T.

NIP: 19760809 2001122001

Pembimbing 2)



**SURABAYA
FEBRUARI 2014**

[Halaman ini sengaja dikosongkan]

Rancang Bangun Permainan Waktu Nyata “Yuk Kita Cari” dengan Banyak Pemain Menggunakan Flash ActionScript 3 (AS3) dan SmartFoxServer2X (SFS2X)

Nama Mahasiswa : Mohammad F K Umam
NRP : 5106 100 064
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing : Imam Kuswardayan S.Kom., M.T.
Sarwosri S.Kom., M.T.

ABSTRAK

Unsur-unsur kegembiraan dan kompetisi yang terdapat dalam sebuah permainan dapat dimanfaatkan untuk mendukung pendidikan. Berbagai permainan telah dikembangkan untuk membantu anak belajar bahasa asing atau mengenal kosakata bahasa asing tersebut dalam suasana kompetisi yang menyenangkan. Namun, tidak banyak dari permainan-permainan tersebut yang melibatkan interaksi antar jaringan sehingga para pemain bisa bermain bersama dan saling berkompetisi secara daring. Padahal, sebuah permainan akan lebih menarik jika melibatkan interaksi dengan pemain lain. Oleh karena itu, dibutuhkan sebuah permainan waktu nyata dengan banyak pemain berbasis web yang dapat membantu memperkaya kosakata bahasa asing bagi anak-anak. Tugas Akhir ini bertujuan untuk membangun permainan “Yuk Kita Cari” yang diharapkan dapat menginisiasi pemenuhan kebutuhan tersebut.

Teknologi yang digunakan adalah Flash ActionScript 3 dan SmartFoxServer2X. Platform Flash digunakan untuk membuat sisi aplikasi klien yang berupa jendela-jendela antarmuka tampilan permainan. Agar dapat dimainkan oleh

banyak pemain, aplikasi klien satu dengan yang lain disinkronisasikan dengan menggunakan bantuan SmartFoxServer2X. SmartFoxServer2X juga berguna untuk menjembatani koneksi antara aplikasi klien dengan basis data. SmartFoxServer2X menyesuaikan logika ketika berkomunikasi dengan aplikasi klien dengan menggunakan sebuah ekstensi Java.

Pengujian dilakukan dengan dua pemain pada lokasi permainan "teras". Dari pengujian dapat disimpulkan bahwa aplikasi klien memiliki performa stabil ketika dimainkan, rata-rata 24 fps (frame per second). SmartFoxServer2X berhasil menangani sinkronisasi banyak klien dengan kecepatan transfer data antara 0,49 KB/detik hingga 0,69 KB/detik. Ke depan, diharapkan permainan "Yuk Kita Cari" terus dikembangkan sehingga dapat memberikan manfaat lebih untuk membantu memperkaya kosakata bahasa asing bagi anak-anak.

Kata kunci: Bahasa asing, Flash ActionScript 3, Pendidikan, Permainan dengan banyak pemain, SmartFoxServer2X

Design and Implementation of Real-time Multiplayer Game “Yuk Kita Cari” Using Flash ActionScript 3 (AS3) and SmartFoxServer2X (SFS2X)

Student Name : Mohammad F K Umam
NRP : 5106 100 064
Major : Teknik Informatika FTIf-ITS
Advisor : Imam Kuswardayan S.Kom., M.T.
Sarwosri S.Kom.,M.T.

ABSTRACT

The excitement and competition elements that can be found at a game can be used to support education. Many various games have been developed to help a children to learn foreign language or to enrich the vocabulary of that foreign language. But, not many of those game that involve networking as it's interaction system so that all player can play together and compete each other online. Whereas a game will be more interesting if it involves interaction between player. Thus, the web-based real-time multiplayer game to help children to enrich the vocabulary of foreign language is required. This final project aims to create “Yuk Kita Cari” game for initiating the accomplishment of that requirement .

The technology used in this final project are Flash ActionScript 3 and SmartFoxServer2X. Flash platform is used to create client application that composed by display interfaces of the game. The client application is synchronized with each other using SmartFoxServer2X assistance, so it can be played by many players. SmartFoxServer2X also useful to bridge the connection between the client application with the database. Java extensions

used by SmartFoxServer2X for adapting logic when communicating with the client application.

Test carried out by 50 player. From the test can be concluded that the client application has a stable performance when played, an average of 24 fps (frames per second). SmartFoxServer2X successfully handle many clients synchronization with data transfer speeds between 0,49 KB/sec up to 0,69 KB/sec. In the future, we hope “Yuk Kita Cari” game can be upgraded so it can provide more benefit to the children to enrich the vocabulary of foreign language.

Keywords: Education, Flash ActionScript 3, Foreign language, Multiplayer game, SmartFoxServer2X

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT, atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

***Rancang Bangun Permainan Waktu Nyata “Yuk Kita Cari”
dengan Banyak Pemain Menggunakan Flash ActionScript 3
(AS3) dan SmartFoxServer2X (SFS2X)***

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Ayah dan Ibu yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Bapak Imam Kuswardayan dan Ibu Sarwosri selaku dosen pembimbing.
3. Seluruh Ustadz, khususnya Ustadz Syaiful, Ustadz Alfin, Ustadz Hisyam Hidayat, dan Ustadz Faiq Furqon.
4. Bayu, Galih, Jawwad, dan teman-teman kontrakan GRS22 dan Marine66 lainnya.
5. Teman-teman *game developer*, khususnya Afif dan Rindra..
6. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Segala kritik dan saran sangat penulis harapkan sebagai bahan perbaikan ke depan.

Surabaya, Februari 2014

Mohammad F K Umam

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
DAFTAR KODE SUMBER	xxv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	3
1.3. Rumusan Permasalahan	3
1.4. Batasan Permasalahan	3
1.5. Metodologi	4
1.6. Sistematika Penulisan.....	6
BAB II KAJIAN PUSTAKA	9
2.1. Penelitian Terkait	9
2.2. Permainan.....	12
2.3. Permainan Waktu Nyata dengan Banyak Pemain.....	13
2.4. Flash ActionScript 3 (AS3).....	14
2.5. SmartFoxServer2X (SFS2X)	14
BAB III ANALISIS DAN PERANCANGAN.....	17
3.1 Analisis.....	17
3.1.1 Ranah Permasalahan dan Deskripsi Umum	17
3.1.2 Arsitektur Perangkat Lunak	18
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak	19
3.2 Analisis Kebutuhan Fungsional	20
3.2.1 <i>Use Case</i>	21
3.2.2 Spesifikasi <i>Use Case</i>	21
3.2.2.1 <i>Use Case</i> Menentukan Lokasi Permainan.....	21
3.2.2.2 <i>Use Case</i> Bergabung ke Dalam Ruang	22
3.2.2.3 <i>Use Case</i> Memulai Permainan	23
3.2.2.4 <i>Use Case</i> Berbincang-bincang	24

3.2.2.5	<i>Use Case</i> Mencocokkan Benda.....	30
3.2.2.6	<i>Use Case</i> Menggunakan Bantuan	31
3.3	Perancangan Perangkat Lunak	33
3.3.1	Perancangan Aplikasi Klien.....	35
3.3.1.1	Perancangan Diagram Kelas	35
3.3.1.1.1	Diagram Kelas Playgame	36
3.3.1.1.2	Diagram Kelas BgWithUnit.....	37
3.3.1.1.3	Diagram Kelas GameplayChatbox.....	38
3.3.1.1.4	Diagram Kelas TimerCountDown	39
3.3.2	Perancangan Ekstensi SmartFoxServer2X.....	42
3.3.3	Perancangan Algoritma.....	44
3.3.3.1	Algoritma Pendaftaran	45
3.3.3.2	Algoritma Permainan	45
3.3.3.3	Algoritma Pencocokan Benda.....	45
3.3.3.4	Algoritma Pemeriksaan Pemenang	46
3.3.3.5	Algoritma Perbincangan Antar Pemain	50
3.3.4	Perancangan Manajemen Data Pengguna	52
3.3.4.1	Perancangan Struktur Basis Data Pengguna	52
3.3.5	Perancangan Tampilan Antarmuka	53
3.3.5.1	Perancangan Entitas Permainan	53
3.3.5.2	Perancangan Peta Permainan	54
3.3.5.3	Perancangan Jendela Permainan	56
BAB IV IMPLEMENTASI.....		63
4.1.	Lingkungan Implementasi.....	63
4.2.	Implementasi Antarmuka	64
4.2.1.	Jendela Menu Utama.....	64
4.2.2.	Jendela “Lobby”	64
4.2.3.	Jendela Pemilihan Peta Permainan	65
4.2.4.	Jendela Ruang Tunggu.....	65
4.2.5.	Jendela Permainan	65
4.2.6.	Implementasi Pendaftaran.....	65
4.2.7.	Implementasi Pembaruan Kata Kunci.....	66
BAB V PENGUJIAN DAN EVALUASI		71
5.1	Lingkungan Pelaksanaan Pengujian.....	71
5.2	Dasar Pengujian	71

5.3 Skenario Pengujian.....	72
5.3.1 Pengujian Fungsional.....	72
5.3.1.1 Pengujian Menentukan Lokasi Permainan.....	72
5.3.1.2 Pengujian Bergabung ke Dalam Ruang	74
5.3.1.3 Pengujian Memulai Permainan	75
5.3.1.4 Pengujian Fitur Perbincangan	78
5.3.1.5 Pengujian Mencocokkan Benda.....	79
5.3.2 Pengujian Performa.....	81
5.3.2.1 Pengujian Performa Klien (UCT006)	81
5.3.2.2 Pengujian Performa Server	84
5.3.2.2.1 Pengujian Performa Saat Klien Dalam Kondisi Diam (UCT007).....	84
5.3.2.2.2 Pengujian Performa Saat Klien Melakukan Perbincangan (UCT008).....	86
5.3.2.2.3 Pengujian Performa Saat Klien Membuat Ruangan (UCT009).....	86
5.4 Evaluasi Pengujian.....	88
BAB VI KESIMPULAN DAN SARAN.....	91
6.1 Kesimpulan	91
6.2 Saran.....	91
DAFTAR PUSTAKA.....	93
LAMPIRAN A Diagram Kelas Permainan “Yuk Kita Cari”	95
LAMPIRAN B Kode Sumber Register.php.....	101
LAMPIRAN C Kode Sumber Forgotpassword.php.....	107
LAMPIRAN D Kode Sumber Pengecekan	113
LAMPIRAN E Kode Sumber Aplikasi Klien	117
BIODATA PENULIS.....	121

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Perbandingan Permainan “Yuk Kita Cari” dengan Permainan Lainnya.....	12
Tabel 3.1 Spesifikasi <i>Use Case</i> Menentukan Lokasi Permainan	22
Tabel 3.2 Spesifikasi <i>Use Case</i> Bergabung ke Dalam Ruang.....	25
Tabel 3.3 Spesifikasi <i>Use Case</i> Memulai Permainan.....	28
Tabel 3.4 Spesifikasi <i>Use Case</i> Berbincang-bincang.....	29
Tabel 3.5 Spesifikasi <i>Use Case</i> Mencocokkan Benda	32
Tabel 3.6 Spesifikasi <i>Use Case</i> Menggunakan Bantuan	34
Tabel 3.7 Daftar Kelas-Kelas Ekstensi SFS2X	42
Tabel 3.8 Daftar Benda dan Kosakata Bahasa Inggris	53
Tabel 3.9 Contoh Permainan Dengan Batasan Maksimal 4 Pemain	56
Tabel 3.10 Keterangan Rancangan Antarmuka.....	58
Tabel 5.1 Prosedur Pengujian Proses Menentukan Lokasi Permainan.....	72
Tabel 5.2 Prosedur Pengujian Proses Bergabung ke Dalam Ruang	74
Tabel 5.3 Prosedur Pengujian Proses Memulai Permainan.....	77
Tabel 5.4 Prosedur Pengujian Proses Memulai Permainan.....	78
Tabel 5.5 Prosedur Pengujian Proses Mencocokkan Benda.....	79

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Tampilan Level 1 Permainan Bahasa Arab Berbasis Desktop.....	10
Gambar 2.2 Pesawat Induk, Jalur, dan Gerbang pada Permainan Solus Saga	11
Gambar 3.1 Arsitektur Sistem	20
Gambar 3.2 Diagram <i>Use Case</i> Pengaturan Ruang.....	22
Gambar 3.3 Diagram <i>Use Case</i> Permainan	24
Gambar 3.4 Diagram <i>Activity</i> Proses Menentukan Lokasi Permainan.....	25
Gambar 3.5 Diagram <i>Activity</i> Proses Bergabung ke Dalam Ruang	27
Gambar 3.6 Diagram <i>Activity</i> Proses Memulai Permainan	29
Gambar 3.7 Diagram <i>Activity</i> Proses Berbincang-bincang	31
Gambar 3.8 Diagram <i>Activity</i> Proses Mencocokkan Benda.....	34
Gambar 3.9 Diagram <i>Activity</i> Proses Menggunakan Bantuan	37
Gambar 3.10 Diagram Kelas Aplikasi Klien.....	38
Gambar 3.11 Diagram Kelas Playgame	40
Gambar 3.12 Diagram Kelas BgWithUnit	41
Gambar 3.13 Diagram Kelas GameplayChatbox	41
Gambar 3.14 Diagram Kelas TimerCountDown	42
Gambar 3.15 Diagram Hubungan Antarkelas Ekstensi SFS2X ..	46
Gambar 3.16 <i>Flowchart</i> Pendaftaran Pengguna.....	47
Gambar 3.17 <i>Flowchart</i> Permainan Secara Umum.....	48
Gambar 3.18 <i>Flowchart</i> Pencocokan Benda	49
Gambar 3.19 <i>Flowchart</i> Pemeriksaan Pemenang	50
Gambar 3.20 <i>Flowchart</i> Perbincangan Antar Pemain.....	51
Gambar 3.21 Struktur Basis Data Permainan “Yuk Kita Cari” ..	52
Gambar 3.22 Rancangan Tampilan Peta Permainan	55
Gambar 3.23 Alur Perpindahan Antar Jendela.....	57
Gambar 3.24 Rancangan Jendela Menu Utama	58
Gambar 3.25 Rancangan Jendela “Lobby”	59
Gambar 3.26 Rancangan Jendela Pemilihan Peta	60
Gambar 3.27 Rancangan Jendela Ruang Tunggu.....	61

Gambar 3.28 Rancangan Jendela Permainan	62
Gambar 4.1 Implementasi Antarmuka Menu Utama	64
Gambar 4.2 Implementasi Jendela “Lobby”	67
Gambar 4.3 Implementasi Jendela Pemilihan Peta Permainan ...	67
Gambar 4.4 Implementasi Jendela Ruang Tunggu.....	68
Gambar 4.5 Implementasi Jendela Permainan	68
Gambar 4.6 Tampilan Halaman Pendaftaran & Halaman Lupa Kata Kunci.....	69
Gambar 5.1 Tampilan Pemain Pembuat Jika Berhasil Menentukan Lokasi Permainan.....	73
Gambar 5.2 Tampilan Pemain Lain Jika Pemain Pembuat Telah Menentukan Lokasi Permainan	74
Gambar 5.3 Tampilan Ruang Tunggu pada Pemain Pembuat.....	76
Gambar 5.4 Tampilan Ruang Tunggu pada Pemain Biasa.....	76
Gambar 5.5 Tampilan Antarmuka Permainan pada Semua Pemain	77
Gambar 5.6 Tampilan Perbincangan Antar Pemain	79
Gambar 5.7 Tampilan Setelah Mencocokkan Benda	80
Gambar 5.8 Tampilan Pengukuran Performa Klien pada Halaman Log Masuk.....	82
Gambar 5.9 Tampilan Pengukuran Performa Klien pada Halaman Lobby.....	82
Gambar 5.10 Tampilan Pengukuran Performa Klien pada Halaman Memilih Lokasi Permainan.....	83
Gambar 5.11 Tampilan Pengukuran Performa Klien pada Halaman Ruang Tunggu Pemain.....	83
Gambar 5.12 Tampilan Pengukuran Performa Klien pada Halaman Permainan	84
Gambar 5.13 Tampilan Salah Satu Klien Ketika 50 Klien Terhubung ke Dalam Server.....	85
Gambar 5.14 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Dalam Kondisi Diam	85
Gambar 5.15 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Saling Berbincang.....	86

Gambar 5.16 Tampilan Klien ke-51 Ketika 50 Klien Lainnya Membuat Ruang	87
Gambar 5.17 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Membuat Ruang	87
Gambar A.1 Diagram Kelas MenuMain	95
Gambar A.2 Diagram Kelas MenuRoom	96
Gambar A.3 Diagram Kelas MenuSelectMap	97
Gambar A.4 Diagram Kelas RoomListItem	98
Gambar A.5 Diagram Kelas TimeKeeper	99
Gambar A.6 Diagram Kelas PlayerData	100

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Permainan merupakan salah satu buah pemikiran manusia. Permainan telah muncul di tengah-tengah kehidupan manusia sejak zaman purba dalam bentuk yang sederhana. Pada tiap masa, selalu muncul permainan baru dengan teknologi yang bersesuaian dengan sarana-sarana yang tersedia saat itu. Sebagai contoh permainan catur, dibuat dengan bentuk papan dan ukiran bidak yang merupakan bentuk fisik teknologi pada masa India kuno. Sebuah permainan dibuat untuk memenuhi dorongan naluri menunjukkan eksistensi diri yang secara alami merupakan bagian dari potensi manusia. Hal itulah yang mendorong munculnya suasana kompetisi, yaitu keberadaan status kalah dan menang. Tentu saja setiap pemain didorong untuk meraih kemenangan. Terlepas dari bagaimana pun cara untuk meraihnya, pemain tetap harus mengikuti aturan main dan tidak diperbolehkan melanggar batasan-batasan yang telah ditetapkan dalam sebuah permainan.

Unsur-unsur kegembiraan dan kompetisi yang terdapat dalam sebuah permainan dapat dimanfaatkan untuk mendukung pendidikan. Berbagai permainan mulai dikembangkan untuk membantu anak belajar bahasa asing atau mengenal kosakata bahasa dalam suasana kompetisi yang menyenangkan. Salah satunya adalah “Petualangan Berburu Bintang Hijaiyah” [1]. Terdapat 8 misi dalam permainan tersebut. Pada misi ke-5, pemain menyusuri sungai katak dan diminta mengumpulkan nama-nama benda berbahasa Arab sesuai dengan gambar yang ditampilkan dalam waktu 60 detik. Peluang yang diberikan sebanyak 3 dan akan berkurang 1 jika salah, namun jika jawaban

benar maka nilai akan bertambah 15 poin. Misi gagal jika peluang telah nol atau waktu telah habis. Jika pemain berhasil menyelesaikan misi, maka pemain mendapatkan bonus 3 bintang. Permainan ini dibuat untuk anak-anak berusia 6 tahun ke atas menggunakan Adobe Flash CS3 Profesional.

Sayangnya permainan "Petualangan Berburu Bintang Hijaiyah" hanya bisa dinikmati oleh seorang pemain saja. Permainan tersebut belum mampu mengakomodasi banyak pemain sekaligus. Permainan itu juga tidak memungkinkan satu pemain berkomunikasi dan berkompetisi secara langsung dengan pemain yang lain. Tentunya ini mempengaruhi kualitas edukasi. Pemain menjadi kurang bersemangat untuk mengulangi permainan ketika semua misi telah terlampaui. Padahal ketika anak-anak menghafal dan memahami bahasa asing, mereka selalu membutuhkan upaya yang berulang-ulang. Pengulangan dan pembiasaan merupakan sebuah keniscayaan bagi siapa pun yang mempelajari bahasa apapun, termasuk bahasa Arab dan Inggris. Hal inilah yang harus dipecahkan, yakni bagaimana merancang sebuah permainan dengan konten edukasi yang memungkinkan pemain berkomunikasi dan berkompetisi dengan pemain-pemain lain sehingga tidak cepat bosan. Kualitas edukasi permainan diharapkan semakin meningkat sehingga target globalnya tercapai, yakni membantu pemain dalam memperkaya kosakata bahasa Arab dan Inggris.

Pada masa sekarang, masalah di atas dapat dipecahkan dengan merancang dan membangun permainan berbasis web. Permainan berbasis web semakin diminati dan banyak dikembangkan. Pesatnya perkembangan permainan berbasis web ditandai dengan banyaknya permainan baru dan portal permainan baru yang bermunculan di internet. Selain itu industri di bidang permainan berbasis web juga menjadi salah satu jenis industri yang berkembang dengan pesat, misalnya Zynga dengan "Farmville" yang mampu menyedot 33 juta pemain dalam dua bulan [2]. Ada juga "Solus Saga" yang tidak dikembangkan oleh industri komersial, namun dikembangkan oleh akademisi [3].

Permainan dengan banyak pemain, merupakan salah satu platform permainan yang digemari. Eksistensi permainan dengan banyak pemain terus berkembang dari masa ke masa. Dengan permainan yang dapat dimainkan dengan banyak pemain, pemain bisa berinteraksi dengan orang lain sehingga permainan yang mengandung konten edukasi menjadi semakin menarik untuk dimainkan. Permainan “Yuk Kita Cari” yang akan dibuat diharapkan mampu menghadirkan solusi atas permasalahan yang telah disebutkan di atas.

1.2. Tujuan

Tujuan dari tugas akhir ini adalah untuk menghasilkan sebuah aplikasi permainan berbasis web yang mengandung konten edukasi yang dapat dimainkan oleh banyak pemain secara bersamaan pada satu waktu.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana merancang aturan permainan dari permainan waktu nyata dengan banyak pemain yang diberi nama “Yuk Kita Cari”
2. Bagaimana mengimplementasikan aturan permainan menggunakan Flash ActionScript 3, kemudian mengintegrasikannya dengan SmartFoxServer2X.
3. Apakah aplikasi klien memiliki performa yang baik untuk dimainkan.
4. Apakah server mampu menangani banyak klien yang terhubung.

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. SmartFoxServer yang digunakan adalah versi gratis dari SmartFoxServer yaitu SmartFoxServer2X Community Edition. Versi SmartFoxServer ini mempunyai fitur yang sama dengan versi berbayar, namun dibatasi hanya dapat menampung 100 pengguna yang terhubung dengan server secara bersamaan.
2. Dalam permainan ini pemain dapat membuat ruang-ruang permainan, sebuah ruang permainan dibatasi hanya dapat menampung 4 pengguna.
3. Dalam permainan ini tidak ada peningkatan level, namun pemain dapat memilih beberapa peta lokasi permainan yang menyajikan tantangan dan batas jumlah pemain yang berbeda.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Analisis kebutuhan dan studi literatur

Mengumpulkan dan menggali informasi dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang akan dibangun. Literatur yang digunakan adalah sebagai berikut:

- a. Teknik pemrograman untuk pengembangan aplikasi berbasis web permainan waktu-nyata menggunakan Flash AS3.
- b. Penggunaan SmartFoxServer2X sebagai server permainan dengan banyak pemain dan integrasi komunikasi dengan aplikasi klien yang berbasis Flash AS3.
- c. Pembuatan ekstensi dari SFS2X yang berbasis Java untuk menyesuaikan dengan sistem permainan.

2. Perancangan sistem

Pada tahap ini dilakukan analisa awal dan mendefinisikan kebutuhan sistem untuk mengetahui masalah yang sedang dihadapi. Dari proses tersebut selanjutnya dirumuskan

rancangan sistem yang dapat memberi pemecahan masalah tersebut. Langkah-langkah pada tahap ini adalah sebagai berikut:

- a. Analisis aktor yang terlibat didalam sistem.
- b. Perancangan diagram *use case*, yang merupakan analisis kebutuhan pada aplikasi yang akan dibangun.
- c. Analisis kebutuhan non-fungsional.
- d. Perancangan manajemen pengguna.
- e. Perancangan sistem komunikasi klien-server dengan SmartFoxServer dan SmartFoxAPI.
- f. Perancangan aplikasi dan aturan permainan.

3. Pembuatan (implementasi)

Pada tahap ini dilakukan pembuatan perangkat lunak yang merupakan implementasi dari rancangan yang telah dibuat sebelumnya. Perincian tahap ini adalah sebagai berikut:

- a. Implementasi manajemen pengguna.
- b. Implementasi komunikasi klien-server dengan SmartFoxServer dan SmartFoxAPI.
- c. Pembuatan aturan permainan.

4. Pengujian dan evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan skenario yang telah dipersiapkan sebelumnya. Pengujian dan evaluasi perangkat dilakukan untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan. Tahapan-tahapan dari pengujian yang akan dilakukan adalah sebagai berikut:

- a. Pengujian kebutuhan fungsional.
- b. Pengujian performa aplikasi klien dan server.

5. Penyusunan buku tugas akhir

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir. Buku tugas akhir ini bertujuan untuk memberikan gambaran dari pengerjaan tugas akhir ini dan

diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

1.6. Sistematika Penulisan

Pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir. Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

Secara garis besar, buku tugas akhir nantinya terdiri atas beberapa bagian yaitu:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Kajian Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas mengenai desain dan perancangan perangkat lunak. Desain perangkat lunak meliputi desain data, arsitektur, dan proses.

Bab IV Implementasi

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat output yang dihasilkan oleh aplikasi, dan evaluasi untuk mengetahui kemampuan aplikasi.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran-saran untuk pengembangan aplikasi pada masa mendatang.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi kode-kode sumber yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

KAJIAN PUSTAKA

Pada bab ini akan dibahas mengenai kajian pustaka yang menjadi dasar dari pembuatan tugas akhir. Kajian pustaka meliputi analisa terhadap penelitian terkait, konsep permainan, permainan dengan banyak pemain, penggunaan Flash ActionScript 3 (AS3) dalam membangun perangkat lunak, dan penggunaan SmartFoxServer2X (SFS2X) sebagai server permainan.

2.1. Penelitian Terkait

Penelitian terkait permainan edukasi untuk mempermudah belajar bahasa telah banyak dilakukan sebelumnya. Selain "Petualangan Berburu Bintang Hijaiyah" yang telah disebutkan pada bab sebelumnya, ada pula "Game Edukasi Bahasa Arab untuk Anak Berbasis Desktop" [4]. Permainan tersebut menyajikan materi serta cara belajar bahasa Arab dengan mudah, sesuai untuk anak usia 5-8 tahun. Jenis permainan ini adalah petualangan dengan menyelesaikan serangkaian perjalanan yang diperankan oleh seorang tokoh pemain. Dalam permainan ini terdapat 6 level yaitu level 1 dan level 2 (kata benda), 3 (kata sifat), level 4 dan 5 (kata kerja), level 6 (kalimat). Di setiap level, pemain harus menyelesaikan semacam kuis untuk menambah pengetahuan tentang bahasa Arab. Namun, permainan tersebut masih memiliki beberapa kekurangan yakni hanya dapat dimainkan oleh seorang pemain saja dan hanya terfokus pada penguatan kosakata bahasa Arab.

Misalnya saja pada level 1, pemain diminta mencocokkan benda-benda sesuai dengan kosakata bahasa Arabnya. Gambar 2.1 menunjukkan tampilan benda dan nama-nama benda yang akan dicocokkan pada level 1. Model permainan ini mirip dengan permainan "Yuk Kita Cari" yang akan dibuat. Namun permainan "Yuk Kita Cari" memiliki setidaknya dua keunggulan yakni selain dapat menambah kosakata bahasa Arab juga dapat

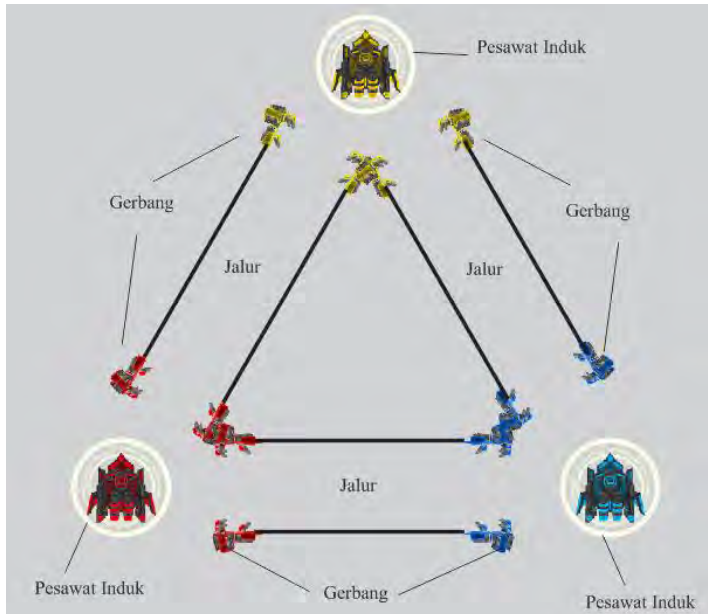
menambah kosakata bahasa Inggris. Di samping itu, permainan “Yuk Kita Cari” dapat dimainkan oleh banyak pemain dalam satu waktu.



Gambar 2.1 Tampilan Level 1 Permainan Bahasa Arab Berbasis Desktop

Penelitian terkait permainan dengan banyak pemain sebenarnya telah banyak dilakukan. Misalnya saja permainan Solus Saga [3]. Permainan ini memang dapat dimainkan oleh banyak pemain dalam satu waktu yang dibangun dengan menggunakan Flash AS3 dan SmartFoxServer2X (SFS2X) sebagai server permainan. Namun, permainan tersebut justru tidak memiliki unsur edukasi sama sekali. Hal ini terlihat pada aturan main dan tujuan permainan. Permainan ini memiliki tujuan untuk menghancurkan semua markas atau kapal induk musuh. Antara pemain satu dengan pemain lainnya terhubung dengan jalur-jalur, seperti terlihat pada Gambar 2.2. Pada setiap ujung jalur tersebut, terdapat gerbang-gerbang yang berguna untuk memanggil

pesawat. Selain untuk memanggil pesawat, gerbang juga digunakan untuk melakukan serangan spesial.



Gambar 2.2 Pesawat Induk, Jalur, dan Gerbang pada Permainan Solus Saga

Permainan “Yuk Kita Cari” yang akan dibangun memiliki perbedaan dibandingkan dengan permainan Solus Saga. Meski sama-sama dibangun dengan menggunakan Flash AS3 dan SmartFoxServer2X (SFS2X) sebagai server permainan, permainan “Yuk Kita Cari” lebih menonjolkan proses edukasi untuk menambah kosakata bahasa Inggris dan Arab bagi setiap pemain. Hal ini tentu menjadi sebuah nilai tambah bagi permainan tersebut karena dengan adanya unsur edukasi, pemain tidak hanya mendapatkan hiburan, namun juga mendapatkan wawasan baru tentang bahasa asing. Apabila dibuat tabel

perbandingan, maka akan nampak keunggulan permainan “Yuk Kita Cari” sebagaimana ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Perbandingan Permainan “Yuk Kita Cari” dengan Permainan Lainnya

Nama Permainan	Unsur Edukasi		Penggunaan Teknologi		Unsur Hiburan
	Bahasa Inggris	Bahasa Arab	Waktu Nyata	Banyak Pemain	
Bahasa Arab untuk Anak Berbasis Desktop	Tidak tersedia	Terse-dia	Tidak tersedia	Tidak tersedia	Tersedia
Petualangan Berburu Bintang Hijaiyah	Tidak tersedia	Terse-dia	Tidak tersedia	Tidak tersedia	Tersedia
Yuk Kita Cari	Terse-dia	Terse-dia	Terse-dia	Terse-dia	Tersedia

2.2. Permainan

Permainan adalah sebuah kegiatan yang bersifat menyenangkan dan hiburan, namun terkadang juga dapat diimbui unsur edukatif didalamnya. Permainan selalu mempunyai aturan dan tujuan, dimana pemain harus menaati aturan-aturan tersebut dan berusaha untuk mencapai tujuan yang telah ditetapkan oleh aturan tersebut.

Sebuah permainan adalah sebuah sistem berbasis aturan formal dengan hasil yang berupa variabel dan dapat diukur, di mana hasil yang berbeda akan mendapatkan nilai yang berbeda, pemain akan berusaha untuk mempengaruhi hasil, dan pemain merasa terikat pada hasilnya, dan konsekuensi dari kegiatan yang opsional dan dapat dinegosiasikan [5].

Permainan sering diklasifikasikan oleh komponen yang diperlukan untuk memainkannya (misalnya miniatur, bola, kartu,

papan dan bidak, atau komputer). Permainan seperti petak umpet tidak memanfaatkan alat apapun. Interaktifitas mereka ditentukan oleh lingkungan.

Permainan dapat dicirikan oleh "apa yang pemain dapat lakukan." Hal ini sering disebut sebagai aturan permainan. Elemen kunci umum yang diidentifikasi dalam konteks ini adalah alat dan aturan yang mendefinisikan konteks keseluruhan permainan dan yang pada gilirannya menghasilkan keterampilan, strategi, dan kesempatan.

Selain diklasifikasikan berdasarkan alat-alat yang digunakan, permainan juga sering ditentukan oleh aturan yang digunakan. Perubahan yang signifikan dalam aturan biasanya menghasilkan permainan baru. Aturan-aturan tersebut pada umumnya menentukan urutan giliran, hak-hak dan tanggung jawab pemain, dan tujuan masing-masing pemain [5].

Kondisi menang yang umum adalah bagaimana pemain yang terlibat menjadi yang pertama untuk mengumpulkan kuota tertentu poin, memiliki jumlah poin terbesar pada akhir permainan, atau beberapa hubungan token permainan seseorang untuk orang lawan seseorang (seperti dalam catur atau skakmat).

Ketika permainan menggunakan alat dan aturan, maka muncul kebutuhan akan keterampilan, strategi, keberuntungan, atau kombinasi keduanya.

2.3. Permainan Waktu Nyata dengan Banyak Pemain

Permainan waktu nyata dengan banyak pemain adalah permainan yang dapat mengakomodasi beberapa pemain untuk bermain dalam sebuah lingkungan permainan yang sama dalam waktu yang sama.

Saat ini permainan dengan banyak pemain masih menjadi salah satu segmen pasar besar pada industri permainan dunia. Hal ini dapat dibuktikan dengan terus bermunculannya banyak permainan dengan banyak pemain dengan jumlah pemain puluhan hingga MMOG (*Massively Multiplayer Online Game*) yang dapat dimainkan ratusan ribu orang secara bersamaan [6].

2.4. Flash ActionScript 3 (AS3)

ActionScript 3.0 adalah bahasa pemrograman berbasis obyek yang menandakan sebuah langkah penting dalam evolusi kemampuan *runtime* FlashPlayer. ActionScript 3.0 didasarkan pada ECMAScript, bahasa pemrograman standar internasional untuk *scripting*. ActionScript 3.0 memperkenalkan ActionScript Virtual Machine yang lebih optimal dibanding versi sebelumnya yaitu AVM2, yang secara dramatis melebihi kinerja dari AVM1. Akibatnya, ActionScript 3.0 mengeksekusi kode hingga 10 kali lebih cepat daripada kode ActionScript versi sebelumnya [7].

Dalam pembuatan permainan “Yuk Kita Cari”, Flash AS3 digunakan untuk membangun aplikasi pada sisi klien. Saat ini, hampir semua peramban mampu mengeksekusi aplikasi Flash AS3 yang berbentuk berkas swf. Maka dari itu, permainan “Yuk Kita Cari” dibangun menggunakan Flash AS3 agar bisa dimainkan secara daring lewat peramban apapun oleh para klien.

2.5. SmartFoxServer2X (SFS2X)

SmartFoxServer2X adalah *multi-platform socket server* yang dirancang untuk diintegrasikan dengan Adobe Flash, memungkinkan pengembang untuk secara cepat mengembangkan aplikasi banyak pengguna dan permainan dengan banyak pemain. SmartFoxServer mendukung semua FlashPlayer mulai dari versi 6 sampai yang terbaru Flash Player 10 dan menyediakan seperangkat API untuk ActionScript 2.0, ActionScript 3.0, dan Java, .Net/Unity3D, Shockwave dan banyak bahasa lainnya. SmartFoxServer2X mendukung hampir semua sistem operasi masa kini, mulai dari sistem operasi berbasis Linux/Unix, Windows maupun Mac [8].

SmartFoxServer2X juga memungkinkan pengembang untuk menciptakan berbagai aplikasi banyak pemain dari perbincangan sederhana dan permainan *turn-based* hingga MMO skala besar dengan pengelompokan.

SmartFoxServer2X mempunyai ekstensibilitas yang tinggi. Fitur SFSExtension memungkinkan pengembang untuk menambahkan logika dalam server. Logika ditambahkan dengan membuat kelas-kelas *handler* yang disatukan dalam sebuah pustaka Java.

SmartFoxServer2X juga mendukung koneksi pada basis data. Hal ini dapat memberi keuntungan kepada beberapa platform yang masih belum bisa mengakses basis data secara langsung. Beberapa platform yang biasa digunakan untuk membangun aplikasi permainan namun belum dapat mengakses basis data secara langsung adalah Flash, Unity, dan aplikasi *mobile*.

Dalam berkomunikasi dengan server SmartFoxServer 2X, aplikasi klien akan menggunakan SmartFoxAPI. SmartFoxAPI akan menangani protokol komunikasi antara klien dan server. SmartFoxAPI saat ini telah dibuat untuk platform Flash, iPhone/iPad, .NetFramework, Unity3D, Silverlight, JavaSE/EE/Android, Director/Shockwave, dan Ajax [8].

Berbagai macam fitur di atas, tentu akan turut memudahkan pembuatan permainan “Yuk Kita Cari”. SmartFoxServer2X digunakan sebagai server untuk melakukan berbagai proses penyalarsan data antar pemain. Proses-proses penyalarsan data tersebut ditangani oleh kelas-kelas *handler* yang disatukan dalam pustaka Java (.jar), kemudian dimuat oleh SmartFoxServer2X ketika aplikasi server tersebut dinyalakan.

Masalah ketidakmampuan Flash AS3 untuk mengakses basis data secara langsung juga dapat dipecahkan melalui penggunaan SmartFoxServer2X. Sebagaimana yang telah disebutkan di atas, SmartFoxServer2X mendukung koneksi secara langsung pada basis data sehingga dalam pembuatan permainan “Yuk Kita Cari” data pemain dari server basis data bisa dimuat melalui SmartFoxServer2X untuk kemudian dikirimkan kepada para pemain. Data yang dikirimkan dari SmartFoxServer2X selanjutnya dimuat oleh aplikasi Flash AS3 yang dijalankan melalui peramban untuk diproses lebih lanjut dalam permainan.

Begitu juga sebaliknya, aplikasi klien yang dibuat dari Flash AS3 juga dapat melakukan penyimpanan data statistik permainan ke dalam server basis data melalui SmartFoxServer2X.

Oleh karenanya, hampir semua fitur dalam permainan “Yuk Kita Cari” ditangani oleh SmartFoxServer2X yang berkolaborasi dengan kelas-kelas *handler* pustaka Java dan kelas-kelas pada aplikasi klien yang dibangun dengan Flash AS3. Fitur-fitur tersebut khususnya yang khususnya yang membutuhkan proses pertukaran dan penyelarasan data semisal perbincangan antar pemain, perbandingan skor, atau pun pencocokan kosakata.

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis sistem, perancangan sistem, perancangan perangkat lunak, dan implementasi perangkat lunak dari permainan “Yuk Kita Cari”.

3.1 Analisis

Tahap analisis ini terbagi menjadi beberapa bagian antara lain: ranah permasalahan dan deskripsi umum perangkat lunak, arsitektur perangkat lunak, dan spesifikasi kebutuhan perangkat lunak. Berikut penjabaran bagian-bagian tahap analisis.

3.1.1 Ranah Permasalahan dan Deskripsi Umum

Ketika anak-anak menghafal dan memahami kosakata bahasa asing, mereka selalu membutuhkan upaya yang berulang-ulang. Pengulangan dan pembiasaan merupakan sebuah keniscayaan bagi siapa pun yang mempelajari bahasa apapun, termasuk bahasa Arab dan Inggris. Hal inilah yang harus dipecahkan, yakni bagaimana merancang sebuah permainan dengan konten edukasi yang memungkinkan pemain berkomunikasi dan berkompetisi dengan pemain-pemain lain sehingga tidak cepat bosan. Kualitas edukasi permainan diharapkan semakin meningkat sehingga target globalnya tercapai, yakni membantu pemain dalam memperkaya kosakata bahasa Arab dan Inggris.

Penelitian terkait permainan edukasi untuk mempermudah belajar bahasa telah banyak dilakukan sebelumnya. Selain “Petualangan Berburu Bintang Hijaiyah” yang telah disebutkan pada bab sebelumnya, ada pula “Game Edukasi Bahasa Arab untuk Anak Berbasis Desktop” [4]. Permainan tersebut menyajikan materi serta cara belajar bahasa Arab dengan mudah, sesuai untuk anak usia 5-8 tahun. Jenis permainan ini

adalah petualangan dengan menyelesaikan serangkaian perjalanan yang diperankan oleh seorang tokoh pemain. Dalam permainan ini terdapat 6 level yaitu level 1 dan level 2 (kata benda), 3 (kata sifat), level 4 dan 5 (kata kerja), level 6 (kalimat). Di setiap level, pemain harus menyelesaikan semacam kuis untuk menambah pengetahuan tentang bahasa Arab. Namun, permainan tersebut masih memiliki beberapa kekurangan yakni hanya dapat dimainkan oleh seorang pemain saja dan hanya terfokus pada penguatan kosakata bahasa Arab.

Misalnya saja pada level 1, pemain diminta mencocokkan benda-benda sesuai dengan kosakata bahasa Arabnya. Model permainan ini memang mirip dengan permainan “Yuk Kita Cari” yang akan dibuat. Namun permainan “Yuk Kita Cari” memiliki setidaknya dua keunggulan yakni selain dapat menambah kosakata bahasa Arab juga dapat menambah kosakata bahasa Inggris. Di samping itu, permainan “Yuk Kita Cari” dapat dimainkan oleh banyak pemain dalam satu waktu.

Permainan yang dibangun nantinya juga harus bisa diakses dengan mudah dalam berbagai kesempatan. Untuk menyelesaikan persoalan tersebut, permainan “Yuk Kita Cari” dibangun berbasis web. Dengan permainan yang berbasis web, anak-anak yang ingin bermain sambil belajar bisa mengakses permainan lewat peramban tanpa harus melakukan instalasi yang rumit.

3.1.2 Arsitektur Perangkat Lunak

Dalam permainan “Yuk Kita Cari”, terdapat tiga bagian sistem yang saling terintegrasi. Tiga bagian sistem tersebut adalah manajemen pengguna yang berbasis PHP, aplikasi permainan sisi klien yang berbasis Flash ActionScript 3, dan SmartFoxServer2X dan ekstensinya yang berbasis Java. Manajemen pengguna dalam hal ini adalah sebuah laman web yang berguna bagi pemain baru yang ingin mendaftar agar dapat memainkan permainan “Yuk Kita Cari”.

Bagian sistem yang kedua adalah aplikasi permainan sisi klien yang berbasis Flash AS3. Bagian sistem ini adalah inti dari permainan “Yuk Kita Cari”. Dalam bagian sistem ini akan diimplementasikan aturan permainan yang dimainkan oleh pemain. Aplikasi sisi klien akan berinteraksi dengan SmartFoxServer2X dengan menggunakan SmartFoxAPI.

Bagian sistem yang ketiga adalah SmartFoxServer2X yang bertugas sebagai jembatan komunikasi antara aplikasi klien satu dan yang lainnya. SmartFoxServer2X juga menangani permintaan koneksi basis data dari aplikasi klien. Agar dapat menyesuaikan logika dengan aplikasi klien, akan dibangun sebuah ekstensi berupa pustaka Java yang akan dipasangkan pada SmartFoxServer. Ketiga bagian sistem tersebut dirangkai menjadi sebuah sistem permainan waktu nyata dengan banyak pemain berbasis web “Yuk Kita Cari” yang tergambar pada bagan arsitektur sistem yang dapat dilihat pada Gambar 3.1.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Pada sistem yang akan dibuat ini, dibutuhkan beberapa fungsi yang dapat membantu proses bisnis dalam sistem. Fungsi-fungsi tersebut antara lain:

1. Menangani Proses Pendaftaran Pemain

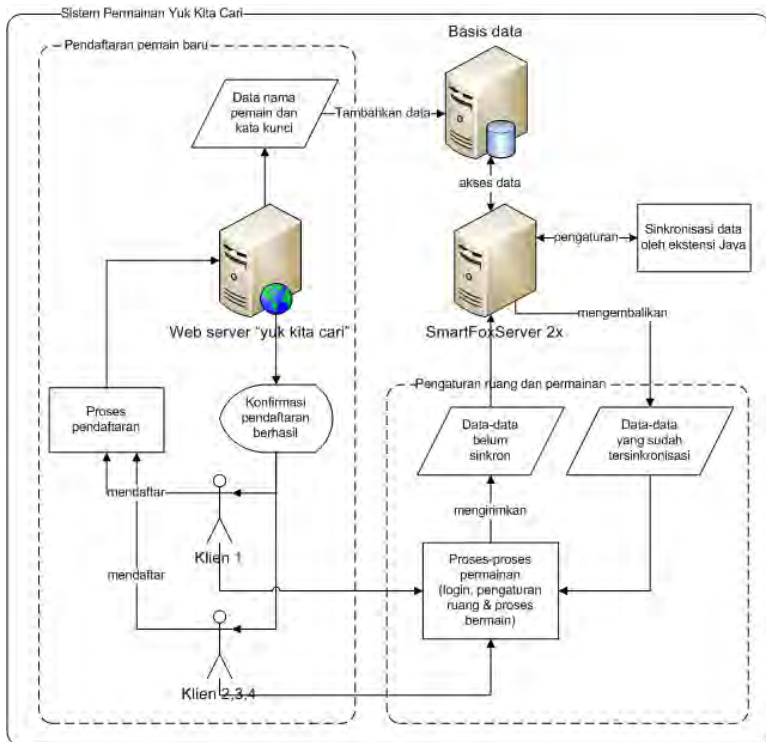
Fungsi ini nantinya akan menyimpan data pemain yang telah dimasukkan pengguna ke dalam antarmuka pendaftaran. Data yang disimpan adalah berupa id pengguna, nama pengguna, kata kunci, pertanyaan rahasia, dan lain-lain.

2. Mengatur Pertukaran Data Antar Pengguna

Fungsi ini nantinya akan menjaga sinkronisasi pertukaran data ketika pengguna berbincang-bincang, memulai membuat peta, bergabung, atau mulai bermain. Data yang dipertukarkan bermacam-macam tergantung kebutuhan.

Bisa berupa id pengguna, nama pengguna, kata-kata, nomor benda-benda, dan lain-lain.

Fungsi-fungsi yang lain dari permainan ini akan dirinci lebih lanjut dalam analisis kebutuhan fungsional dan penjelasan *use case*.



Gambar 3.1 Arsitektur Sistem

3.2 Analisis Kebutuhan Fungsional

Dalam perancangan sistem aplikasi ini, akan dijelaskan mengenai *use case* yang digunakan untuk memberikan gambaran yang lebih jelas mengenai fungsi dan fitur yang terdapat pada

sistem juga desain aturan main yang akan digunakan dalam permainan “Yuk Kita Cari”.

3.2.1 *Use Case*

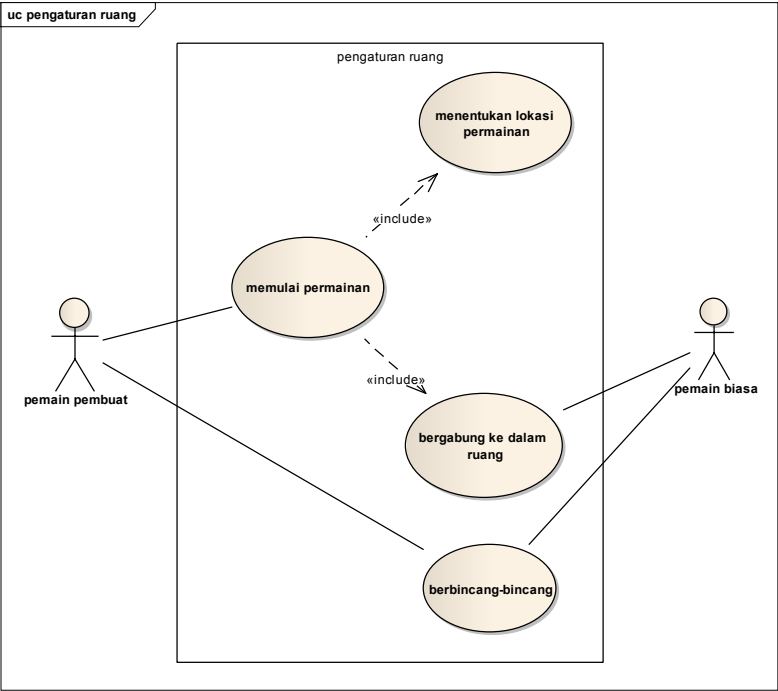
Pada sistem permainan “Yuk Kita Cari”, terdapat dua rancangan diagram *use case*, yakni diagram *use case* pengaturan ruang dan diagram *use case* permainan. Pada *use case* pengaturan ruang, terdapat dua jenis aktor pengguna yaitu aktor pemain pembuat dan aktor pemain biasa. Gambar 3.2 menjelaskan tentang diagram *use case* yang berasosiasi dengan aktor pemain pembuat dan aktor pemain biasa. Aktor pemain pembuat berasosiasi dengan *use case* membuat ruang, menentukan lokasi permainan, bergabung ke dalam ruang, dan berbincang-bincang. Sedangkan aktor pemain biasa berasosiasi dengan *use case* bergabung ke dalam ruang dan berbincang-bincang. Gambar 3.3 menjelaskan tentang diagram *use case* permainan yang berasosiasi dengan pemain secara umum. Dalam permainan, aktor pemain berasosiasi dengan *use case* memilih benda, mencocokkan benda, menggunakan bantuan, keluar dari permainan, dan berbincang-bincang.

3.2.2 *Spesifikasi Use Case*

Pada subbab ini, akan dijelaskan spesifikasi masing-masing *use case* permainan berikut rinciannya dalam diagram *activity*.

3.2.2.1 *Use Case Menentukan Lokasi Permainan*

Dalam *use case* ini, pemain melakukan proses untuk menentukan lokasi permainan. Setiap jenis lokasi memiliki kapasitas ruang dan daftar benda tertentu yang berbeda dengan jenis lainnya. Keterangan lebih lengkap mengenai *use case* menentukan lokasi permainan bisa dilihat dalam Gambar 3.4. Sedangkan skenario *use case* ditunjukkan oleh Tabel 3.1.



Gambar 3.2 Diagram Use Case Pengaturan Ruang

3.2.2.2 Use Case Bergabung ke Dalam Ruang

Dalam *use case* ini, pemain biasa melakukan proses untuk memilih ruang yang sudah ada dan bergabung ke dalamnya. Keterangan lebih lengkap mengenai *use case* bergabung ke dalam ruang bisa dilihat dalam Gambar 3.5. Sedangkan skenario *use case* ditunjukkan oleh Tabel 3.2.

Tabel 3.1 Spesifikasi Use Case Menentukan Lokasi Permainan

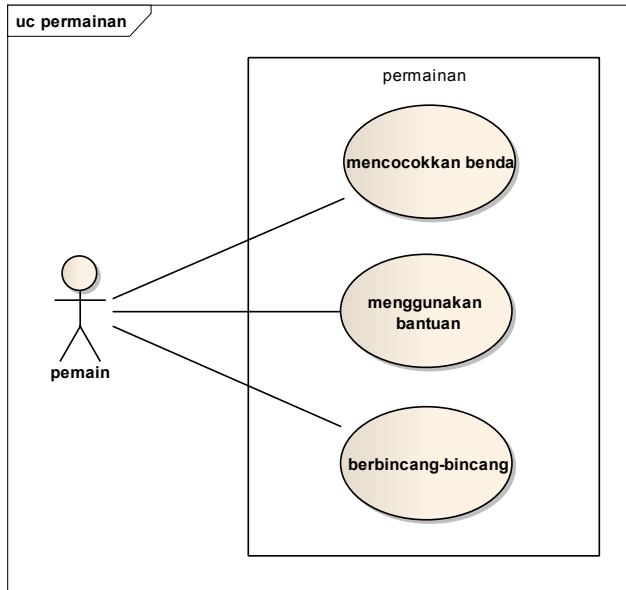
ID Use Case	UC001
Nama Use Case	Menentukan Lokasi Permainan
Aktor:	Pemain Pembuat

Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk menentukan lokasi permainan sebelum memulai permainan.	
Pemicu:	Pemain pembuat menekan tombol “Buat Ruang” pada antarmuka “Lobby”.	
Preconditions:	Pemain pembuat sudah berhasil melakukan log masuk ke dalam antarmuka “Lobby”.	
Postconditions:	1. Ruang baru berhasil dibuat. 2. Pemain pembuat berpindah ke antarmuka “Ruang Tunggu”.	
Alur Normal:	1	Pemain pembuat menekan tombol “Buat Ruang”.
	2	Pemain pembuat masuk ke antarmuka pilihan lokasi permainan.
	3	Pemain pembuat memilih lokasi permainan.
	4	Sistem membentuk ruang baru sesuai dengan lokasi yang telah dipilih pemain pembuat.
	5	Sistem menampilkan antarmuka “Ruang Tunggu”.
Alur Alternatif:	1a	Jika pemain pembuat gagal masuk ke tampilan pilihan peta permainan.
	1a1	Sistem menampilkan antarmuka konfirmasi galat.
	1a2	Kembali ke antarmuka “Lobby”.
Includes:	-	
Asumsi-asumsi:	Pemain pembuat bisa melihat dan membaca dengan baik pada setiap antarmuka.	

3.2.2.3 Use Case Memulai Permainan

Dalam *use case* ini, pemain pembuat melakukan proses untuk memulai permainan. Keterangan lebih lengkap mengenai

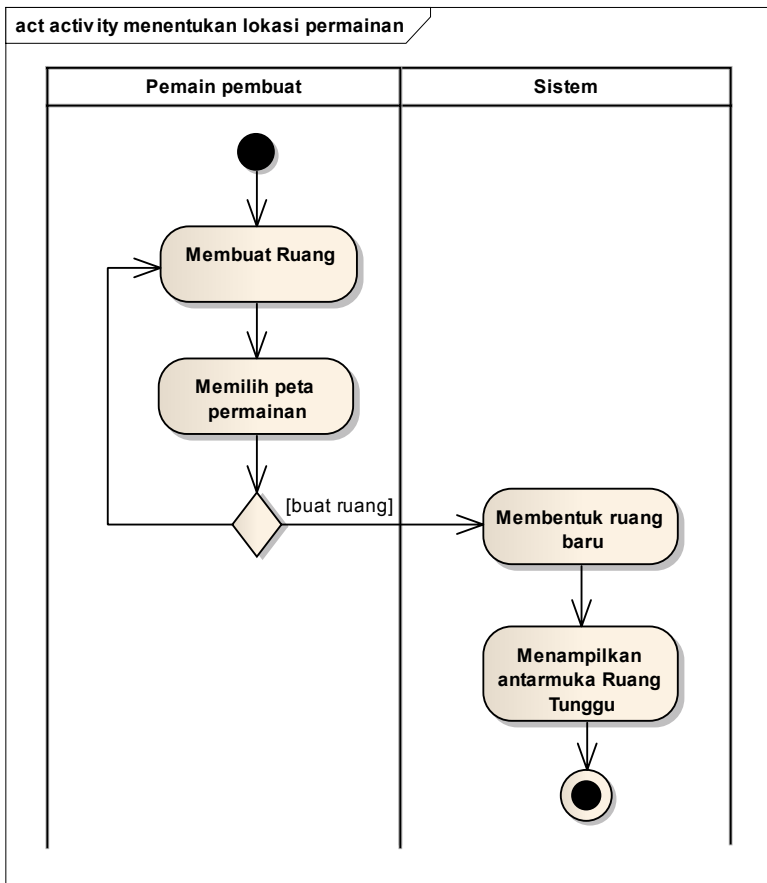
use case memulai permainan bisa dilihat dalam Tabel 3.3 dan Gambar 3.6.



Gambar 3.3 Diagram Use Case Permainan

3.2.2.4 Use Case Berbincang-bincang

Dalam *use case* ini, pemain melakukan proses untuk saling berkomunikasi dengan pemain lainnya. Setiap pemain bisa mengirimkan pesan dan menerima pesan dari pemain lainnya. Gambar 3.7 menunjukkan diagram *activity use case* berbincang-bincang, sedangkan Tabel 3.4 menunjukkan skenario *use case* berbincang-bincang.



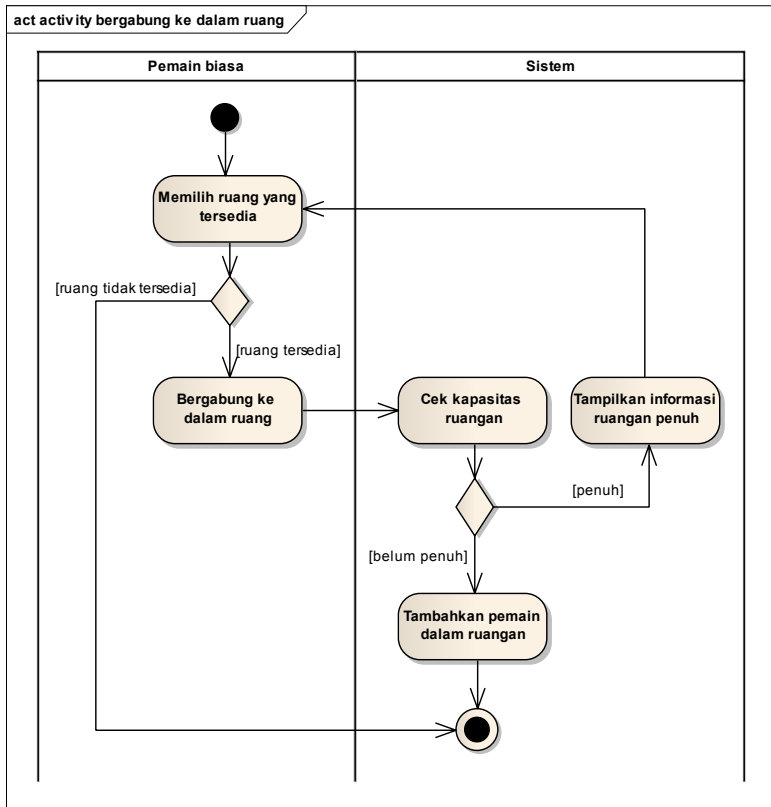
Gambar 3.4 Diagram *Activity* Proses Menentukan Lokasi Permainan

Tabel 3.2 Spesifikasi *Use Case* Bergabung ke Dalam Ruang

ID Use Case	UC002
Nama Use Case	Bergabung ke Dalam Ruang
Aktor:	Pemain Biasa
Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk

	bergabung ke dalam ruang bersama pemain lain sebelum memulai permainan.	
Pemicu:	Pemain biasa menekan tombol “Bergabung ke Dalam Ruang” pada antarmuka “ <i>Lobby</i> ”.	
Preconditions:	1. Pemain biasa sudah berhasil melakukan log masuk ke dalam “ <i>Lobby</i> ”. 2. Ada ruang yang telah dibuat oleh pemain pembuat. 3. Ruang-ruang yang telah ada belum penuh.	
Postconditions:	1. Pemain biasa telah bergabung dalam sebuah ruang yang dibuat oleh pemain pembuat tertentu. 2. Pemain biasa berpindah ke antarmuka “Ruang Tunggu”.	
Alur Normal:	1	Pemain biasa memilih salah satu ruang yang tersedia pada bagian antarmuka “ <i>Lobby</i> ”.
	2	Pemain biasa menekan tombol “Bergabung ke Dalam Ruang”.
	3	Sistem melakukan pemeriksaan terhadap jumlah pemain dalam ruang yang dipilih.
	4	Pemain biasa masuk ke antarmuka “Ruang Tunggu” untuk menunggu pemain pembuat memulai permainan.
Alur Alternatif:	3a	Jika ruang yang dipilih sudah penuh.
	3a1	Tidak bisa masuk ke “Ruang Tunggu”.
	3a2	Sistem menampilkan antarmuka konfirmasi ruang sudah penuh.
	3a3	Kembali ke antarmuka “ <i>Lobby</i> ” untuk memilih ruang lain yang belum

		penuh.
Includes:	-	
Asumsi-asumsi:	Pemain pembuat bisa melihat dan membaca dengan baik pada setiap antarmuka.	

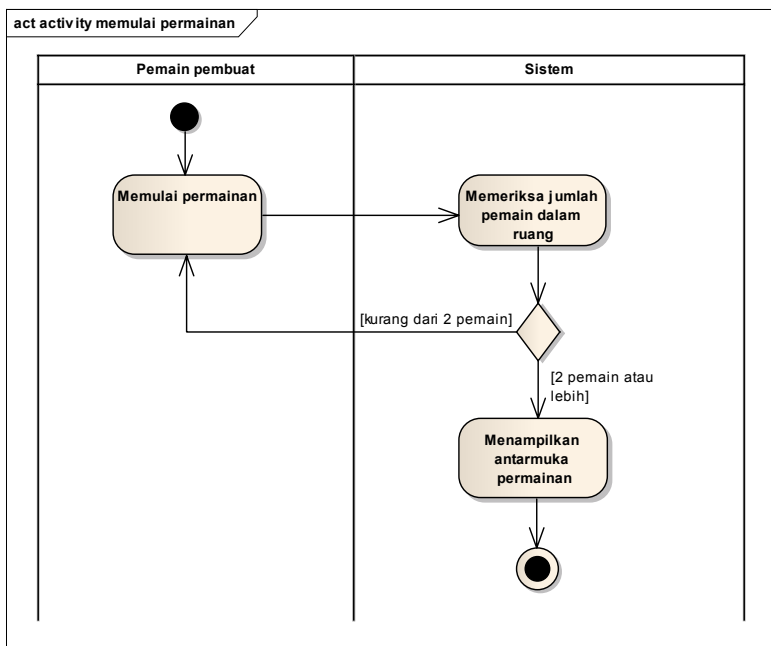


Gambar 3.5 Diagram Activity Proses Bergabung ke Dalam Ruang

Tabel 3.3 Spesifikasi *Use Case* Memulai Permainan

ID Use Case	UC003	
Nama Use Case	Memulai Permainan	
Aktor:	Pemain Pembuat	
Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk memulai permainan.	
Pemicu:	Pemain pembuat menekan tombol “Mulai Permainan” pada antarmuka “Ruang Tunggu”.	
Preconditions:	1. Pemain pembuat sudah berhasil membuat ruang dan masuk ke “Ruang Tunggu”. 2. Ada minimal 2 pemain dalam “Ruang Tunggu”.	
Postconditions:	1. Seluruh pemain dalam “Ruang Tunggu” berpindah masuk ke antarmuka permainan. 2. Pemain bisa bertanding dengan pemain lain.	
Alur Normal:	1	Pemain pembuat menekan tombol “Mulai Permainan” pada antarmuka “Ruang Tunggu”.
	2	Sistem melakukan pemeriksaan jumlah pemain dalam ruang.
	3	Pemain pembuat dan pemain biasa berpindah dari antarmuka “Ruang Tunggu” ke antarmuka permainan.
Alur Alternatif:	2a	Jika pemain pembuat menekan tombol “Mulai Permainan” ketika belum ada pemain lain yang bergabung dalam ruang tersebut.
	2a1	Sistem menampilkan antarmuka konfirmasi tidak bisa bermain sendirian.
	2a2	Kembali ke antarmuka “Ruang Tunggu” untuk menunggu pemain

	lain bergabung.
<i>Includes:</i>	<i>Use case Memilih Lokasi Permainan & use case Bergabung ke Dalam Ruang.</i>
Asumsi-asumsi:	Pemain pembuat bisa melihat dan membaca dengan baik pada setiap antarmuka.



Gambar 3.6 Diagram Activity Proses Memulai Permainan

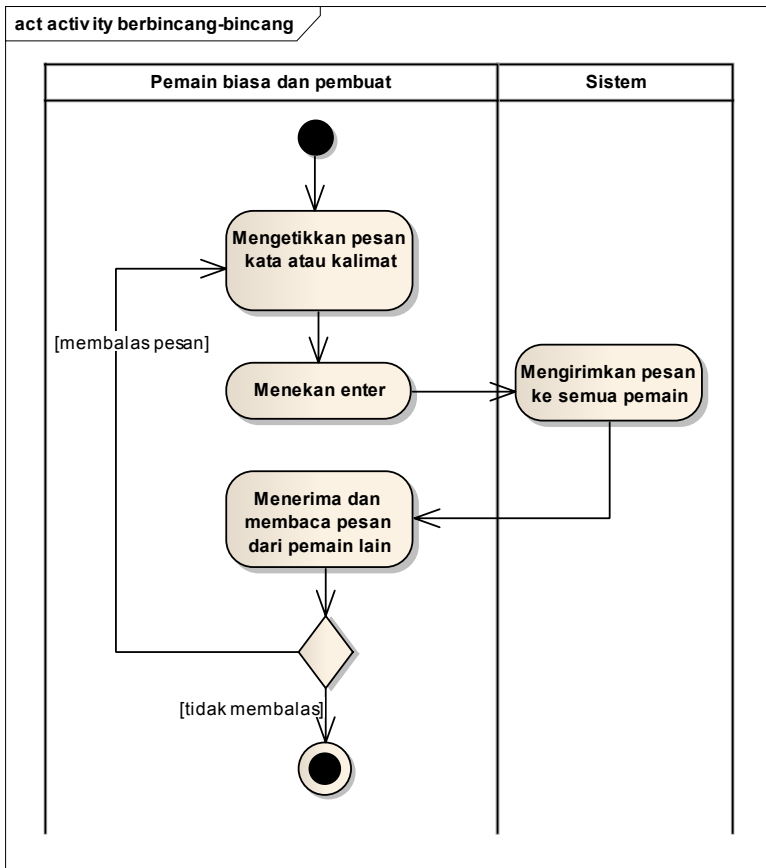
Tabel 3.4 Spesifikasi Use Case Berbincang-bincang

ID Use Case	UC004
Nama Use Case	Berbincang-bincang
Aktor:	Pemain Pembuat, Pemain Biasa
Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk melakukan perbincangan antar pemain.
Pemicu:	Pemain menekan tombol “Enter” setelah

	mengetikkan pesan.	
Preconditions:	Pemain pembuat atau pun biasa sudah berhasil melakukan log masuk ke dalam “Lobby”.	
Postconditions:	1. Kata-kata atau kalimat yang diketikkan pemain terkirim ke pemain-pemain lain. 2. Semua pemain di “Lobby” dapat membaca kata atau kalimat yang dikirim pemain lain sehingga seluruh pemain dapat berbincang-bincang.	
Alur Normal:	1	Pemain mengetikkan pesan kata-kata atau kalimat pada kotak perbincangan.
	2	Pemain menekan tombol “Enter” pada papan tombol.
	3	Sistem mengirimkan pesan kepada seluruh pemain lain.
	2a	Jika pemain menekan tombol “Enter” ketika belum ada pesan kata-kata yang diketikkan dalam kotak perbincangan.
	2a1	Tidak terjadi pengiriman pesan apa pun oleh sistem.
Includes:	-	
Asumsi-asumsi:	Pemain pembuat bisa melihat dan membaca dengan baik pada setiap antarmuka.	

3.2.2.5 Use Case Mencocokkan Benda

Dalam *use case* ini, pemain melakukan proses untuk mencocokkan benda dengan nama benda dalam permainan. Keterangan lebih lengkap mengenai *use case* mencocokkan benda bisa dilihat dalam Tabel 3.5 dan Gambar 3.8.



Gambar 3.7 Diagram Activity Proses Berbincang-bincang

3.2.2.6 Use Case Menggunakan Bantuan

Dalam *use case* ini, pemain melakukan proses untuk menggunakan bantuan sehingga memudahkan dalam mencocokkan benda atau memperpanjang waktu bermain. Keterangan lebih lengkap mengenai *use case* menentukan lokasi permainan bisa dilihat dalam Tabel 3.6 dan Gambar 3.9.

Tabel 3.5 Spesifikasi Use Case Mencocokkan Benda

ID Use Case	UC005	
Nama Use Case	Mencocokkan Benda	
Aktor:	Pemain Pembuat, Pemain Biasa	
Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk melakukan pencocokan benda oleh masing-masing pemain.	
Pemicu:	Pemain memilih dan menekan tombol nama benda, kemudian memilih benda.	
Preconditions:	Pemain pembuat dan pemain biasa sudah masuk ke dalam antarmuka permainan.	
Postconditions:	1. Tombol nama benda yang dipilih berubah warna. 2. Muncul efek sinar atau bayangan pada benda yang dipilih.	
Alur Normal:	1	Pemain memilih dan menekan tombol nama benda.
	2	Sistem menyimpan masukan nama benda yang dipilih.
	3	Pemain memilih benda yang dianggap sesuai dengan nama benda yang sudah dipilih.
	4	Sistem mengirimkan data nama benda yang sudah dicocokkan kepada seluruh pemain.
	5	Sistem menandai benda yang sudah dicocokkan dengan tampilan nama pemain yang sudah mencocokkan.
	6	Sistem membuat benda yang sudah dicocokkan seorang pemain tidak dapat dipilih oleh pemain lainnya.
	7	Sistem menambah skor pemain yang sudah mencocokkan sebuah benda.
Alur Alternatif:	3a	Jika pemain memilih dan menekan tombol nama benda kemudian

		memilih benda yang tidak sesuai namanya.
	3a1	Tidak terjadi pengiriman data apa pun oleh sistem.
	3a2	Sistem mengurangi skor atau jatah waktu pemain.
	3b	Jika pemain menekan tombol nama benda kemudian mengklik tidak pada benda namun di sembarang bagian gambar.
	3b1	Tidak terjadi pengiriman data apa pun oleh sistem.
	3b2	Sistem mengurangi skor atau jatah waktu pemain.
<i>Includes:</i>		-
Asumsi-asumsi:		Pemain pembuat bisa melihat dan membaca dengan baik pada antarmuka permainan.

3.3 Perancangan Perangkat Lunak

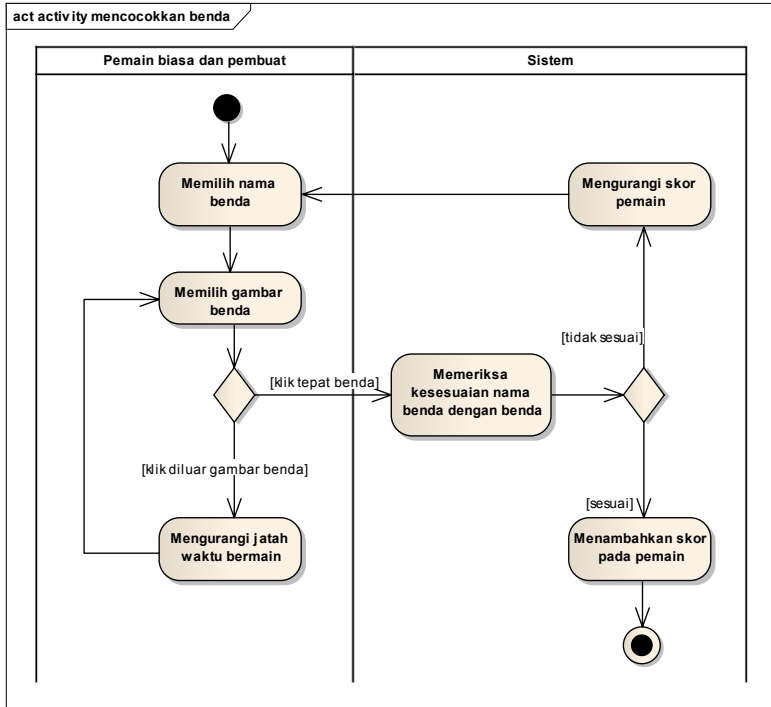
Perancangan perangkat lunak dibagi menjadi beberapa bagian, yaitu perancangan aplikasi dan aturan permainan, perancangan ekstensi dari SmartFoxServer2X, dan perancangan manajemen data pengguna.

Dalam perancangan aturan permainan akan dibahas bagaimana bentuk dari konsep permainan yang terdiri dari aturan permainan, deskripsi unit, dan aksi spesial yang dapat dilakukan oleh pemain.

Selanjutnya, pada perancangan ekstensi SFS2X, akan dibahas bagaimana struktur dan fungsi-fungsi yang dibutuhkan untuk membuat sebuah ekstensi SFS2X sehingga sesuai dengan permainan “Yuk Kita Cari” yang dibuat.

Pada perancangan manajemen data pengguna, akan dibahas mengenai struktur basis data sederhana dan hal-hal yang

harus dilakukan oleh pengguna sebelum dapat memainkan permainan “Yuk Kita Cari”.



Gambar 3.8 Diagram Activity Proses Mencocokkan Benda

Tabel 3.6 Spesifikasi Use Case Menggunakan Bantuan

ID Use Case	UC006
Nama Use Case	Menggunakan Bantuan
Aktor:	Pemain Pembuat, Pemain Biasa
Deskripsi:	Tujuan dari <i>use case</i> ini adalah untuk menggunakan bantuan oleh masing-masing pemain.
Pemicu:	Pemain memilih dan menekan tombol bantuan.

<i>Preconditions:</i>	Pemain pembuat dan pemain biasa sudah masuk ke dalam antarmuka permainan.	
<i>Postconditions:</i>	Tombol nama bantuan yang dipilih berubah warna.	
Alur Normal:	1	Pemain memilih dan menekan tombol nama bantuan.
	2	Sistem menyimpan masukan nama bantuan yang dipilih.
	3	Sistem menjalankan aksi sesuai bantuan yang dipilih pemain.
	4	Bantuan yang telah dipilih tidak dapat lagi digunakan.
Alur Alternatif:	1a	Jika pemain memilih dan menekan tombol bantuan yang sudah pernah dipilih dalam satu permainan.
	1a1	Sistem tidak menjalankan bantuan apa pun.
	1a2	Sistem menampilkan antarmuka konfirmasi bantuan sudah pernah digunakan.
<i>Includes:</i>	-	
Asumsi-asumsi:	Pemain pembuat bisa melihat dan membaca dengan baik pada antarmuka permainan.	

3.3.1 Perancangan Aplikasi Klien

Pada subbab ini akan dibahas perancangan aplikasi klien yang meliputi perancangan diagram kelas, perancangan aturan permainan, dan perancangan fitur berbincang-bincang.

3.3.1.1 Perancangan Diagram Kelas

Diagram kelas pada Gambar 3.10 adalah perancangan kelas-kelas utama pada permainan “Yuk Kita Cari”. Dari diagram kelas tersebut dapat dilihat bahwa kelas MenuLobby berperan

sebagai kontroler utama setelah pengguna melakukan log masuk dari MenuMain.

MenuLobby akan mengatur perpindahan antara jendela satu dengan jendela lain, hal tersebut ditunjukkan dengan asosiasi yang mengarah dari MenuLobby ke kelas-kelas lainnya. Untuk penanganan entitas pada permainan ditangani oleh kelas Playgame yang mempunyai *instance* dari kelas BgWithUnit, dimana kelas BgWithUnit mempunyai *instance* dari kelas Unit.

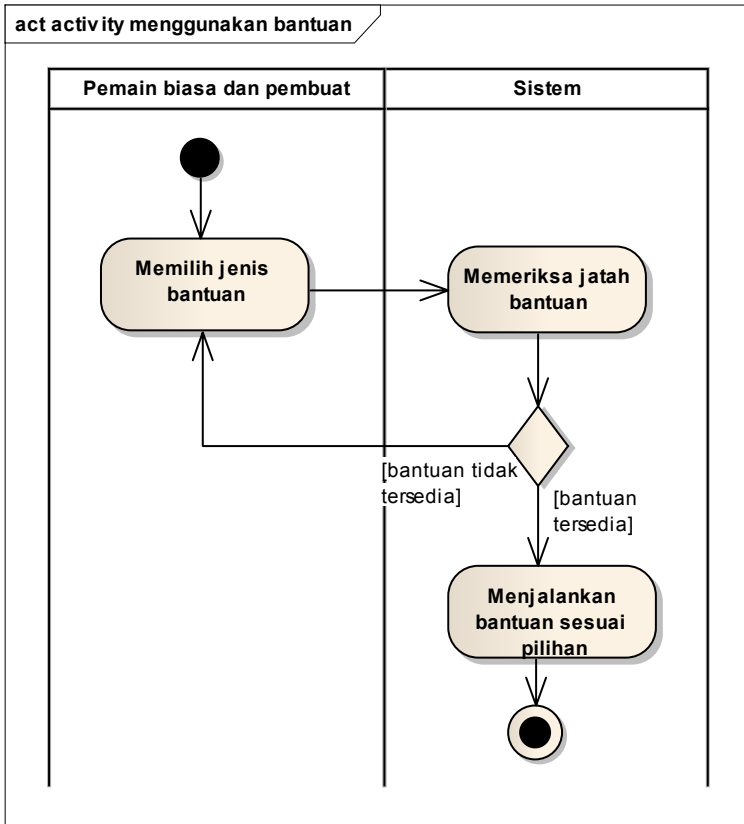
3.3.1.1.1 Diagram Kelas Playgame

Kelas Playgame adalah kelas yang menyimpan informasi berupa aturan permainan dan variabel-variabel dari yang dibutuhkan untuk menjalankan permainan. Kelas ini juga memiliki fungsi untuk melakukan perubahan pada tampilan, berkomunikasi dengan server, dan melakukan manajemen item. Daftar variabel dan fungsi pada kelas Playgame dapat dilihat pada Gambar 3.11.

Pada saat melakukan *update* pada aplikasi, kelas Playgame akan memanggil fungsi *update* yang terdapat pada masing-masing unit, melakukan *update* pada tampilan utama, dan melakukan pengecekan kondisi kemenangan.

Ketika mendapatkan respon dari server, kelas Playgame akan memetakan jenis perintah dari server, kemudian melaksanakan perintah tersebut. Perintah-perintah tersebut berisi proses sinkronisasi antar klien, manajemen benda-benda, dan manajemen pengguna dalam ruangan.

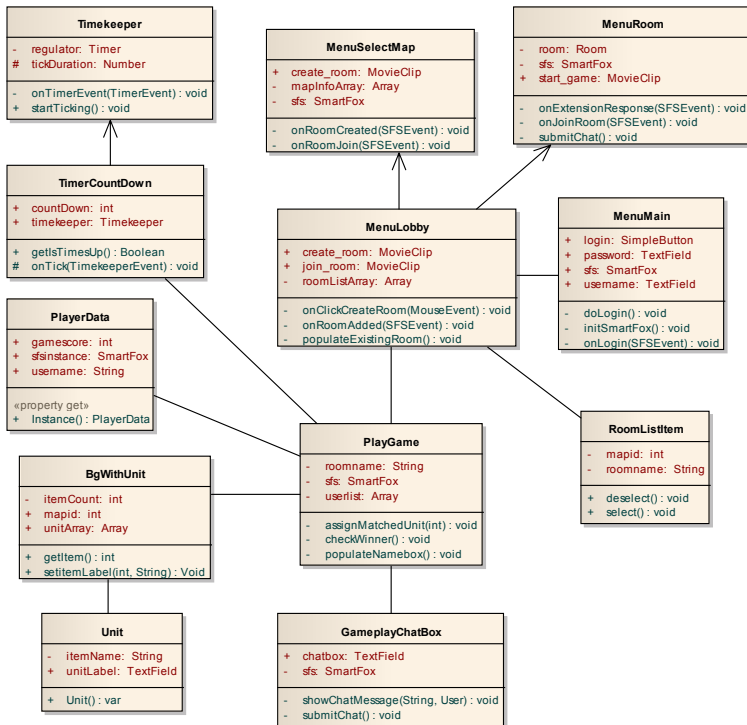
Respon dan permintaan yang ditangani oleh kelas Playgame adalah yang berhubungan dengan pesan permainan dan basis data statistik.



Gambar 3.9 Diagram Activity Proses Menggunakan Bantuan

3.3.1.1.2 Diagram Kelas BgWithUnit

Kelas BgWithUnit menampung objek-objek yang akan dicocokkan dan menggabungkannya dengan tampilan latar belakang permainan. Daftar variabel dan fungsi pada kelas BgWithUnit dapat dilihat pada Gambar 3.12.



Gambar 3.10 Diagram Kelas Aplikasi Klien

3.3.1.1.3 Diagram Kelas GameplayChatbox

Pada permainan “Yuk Kita Cari”, pemain dapat berkomunikasi dengan pemain lain yang berada pada ruangan yang sama dengannya. Fitur perbincangan diimplementasikan oleh kelas MenuLobby, MenuRoom dan kelas GameplayChatbox. Cara kerja kotak perbincangan pada masing-masing kelas tersebut hampir sama. Oleh karena itu kelas yang dipilih untuk dibahas pada subbab ini adalah kelas GameplayChatbox, karena kelas ini adalah kelas yang secara khusus mengimplementasikan fitur

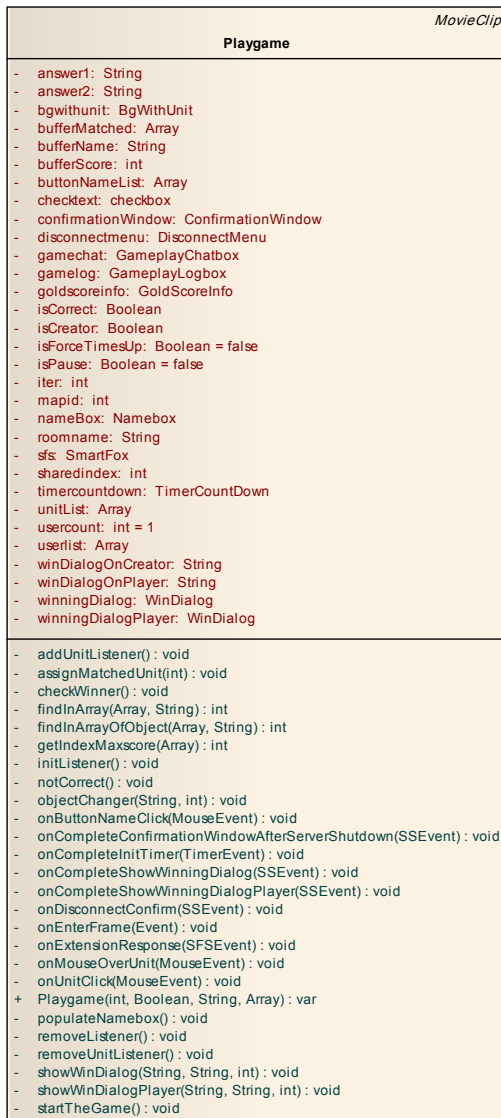
perbincangan. Gambar 3.13 menunjukkan struktur dari kelas `GameplayChatbox`.

Pada saat pemain mengetik pesan dan kemudian menekan tombol “*Enter*” pada baris perbincangan atau menekan tombol “*Enter*” pada papan tombol, maka kelas `GameplayChatbox` akan mengirimkan pesan yang telah diketik oleh pemain kepada server. Selanjutnya server akan menyebarkan pesan tersebut kepada seluruh pemain pada ruang tempat pemain pengirim berada termasuk kepada pemain pengirim sendiri. Hal ini akan memudahkan sinkronisasi pengiriman pesan antar pemain.

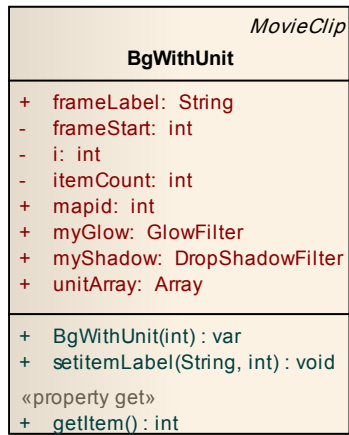
3.3.1.1.4 Diagram Kelas `TimerCountDown`

Saat permainan sedang berlangsung, seluruh pemain dibatasi oleh waktu. Apabila waktu telah habis, maka permainan dihentikan oleh sistem dan sistem segera melakukan perhitungan skor. Namun, kecepatan penghitung waktu bisa berbeda antara satu pemain dengan pemain lainnya tergantung kecepatan eksekusi program oleh komputer. Oleh karena itu kelas yang dipilih untuk dibahas pada subbab ini adalah kelas `TimerCountDown`, karena kelas ini adalah kelas yang secara khusus mengimplementasikan fitur untuk menyelaraskan penghitungan waktu sehingga mampu menghilangkan perbedaan penghitung waktu dalam masing-masing aplikasi klien. Gambar 3.14 menunjukkan struktur dari kelas `TimerCountDown`.

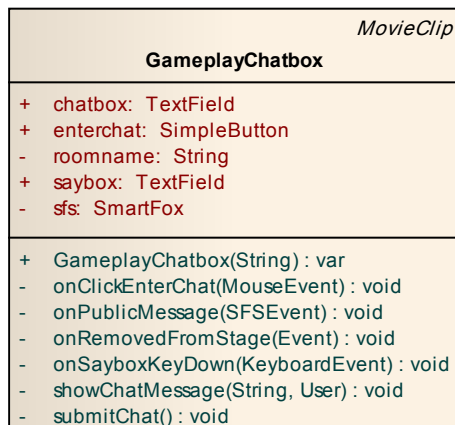
Nampak bahwa kelas `TimerCountDown` mengimplementasikan kelas `Timekeeper`. Kelas tersebut berfungsi untuk mencegah munculnya jarak antar penghitung waktu pada masing-masing klien. Gambar kelas `Timekeeper` dapat dilihat di Lampiran A Gambar A.5.



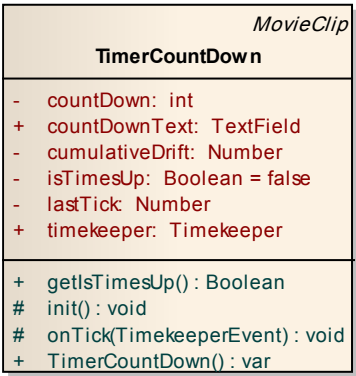
Gambar 3.11 Diagram Kelas Playgame



Gambar 3.12 Diagram Kelas BgWithUnit



Gambar 3.13 Diagram Kelas GameplayChatbox



Gambar 3.14 Diagram Kelas TimerCountDown

3.3.2 Perancangan Ekstensi SmartFoxServer2X

Agar SFS2X dapat menyesuaikan logika dengan aplikasi permainan sisi klien, maka harus dibuat sebuah ekstensi berbasis Java. Ekstensi SFS2X berupa sebuah pustaka Java (.jar) yang berisi kelas-kelas Java.

Kelas-kelas tersebut mengimplementasikan kelas-kelas abstrak *SFSExtension*, *BaseServerEventHandler*, dan *BaseClientRequestHandler* dari pustaka SFS2X. Kelas-kelas ini berisi logika untuk menjembatani komunikasi klien ke klien maupun komunikasi klien ke basis data. Daftar kelas-kelas ekstensi SFS2X dapat dilihat pada Tabel 3.7.

Tabel 3.7 Daftar Kelas-Kelas Ekstensi SFS2X

Nama Kelas	Keterangan
YukariExtension	Kelas abstrak yang diwarisi: <i>SFSExtension</i>

	<p>Kegunaan: Memasangkan kelas-kelas <i>handler</i> kepada server ketika ekstensi dimuat oleh server.</p>
LoginHandler	<p>Kelas abstrak yang diwarisi: <i>BaseServerEventHandler</i></p> <p>Kegunaan: Menangani ketika pengguna melakukan log masuk, kelas ini akan mencocokkan antara nama pengguna dan kata kunci yang dimasukkan dengan data yang tersimpan dalam basis data.</p>
StatsDatabaseHandler	<p>Kelas abstrak yang diwarisi: <i>BaseClientRequestHandler</i></p> <p>Kegunaan: Menangani akses ke basis data untuk keperluan statistik pemain.</p>
CustomDatabaseRequestHandler	<p>Kelas abstrak yang diwarisi: <i>BaseClientRequestHandler</i></p> <p>Kegunaan: Menangani akses ke basis data untuk keperluan umum.</p>
GameMessageHandler	<p>Kelas abstrak yang diwarisi: <i>BaseClientRequestHandler</i></p> <p>Kegunaan:</p>

	Digunakan untuk menangani pesan-pesan komunikasi antar klien pada saat permainan.
CustomMessageHandler	Kelas abstrak yang diwarisi: <i>BaseClientRequestHandler</i> Kegunaan: Digunakan untuk menangani pesan-pesan komunikasi antar klien dengan kegunaan umum.
UserDisconnectHandler	Kelas abstrak yang diwarisi: <i>BaseServerEventHandler</i> Kegunaan: Digunakan untuk menangani proses log keluar dan pemutusan sambungan klien dengan server.

Sedangkan struktur dan hubungan ketujuh kelas tersebut tergambar pada diagram kelas pada Gambar 3.15 Diagram Hubungan Antarkelas Ekstensi SFS2X.

Ketujuh kelas tersebut dibangun menjadi sebuah berkas pustaka Java (.jar) dan dimasukkan ke dalam pelipat “Extension” pada direktori SFS2X.

3.3.3 Perancangan Algoritma

Pada subbab ini akan dibahas beberapa algoritma yang utama dalam proses pembuatan permainan “Yuk Kita Cari”. Algoritma tersebut digambarkan dalam bentuk *flowchart*.

3.3.3.1 Algoritma Pendaftaran

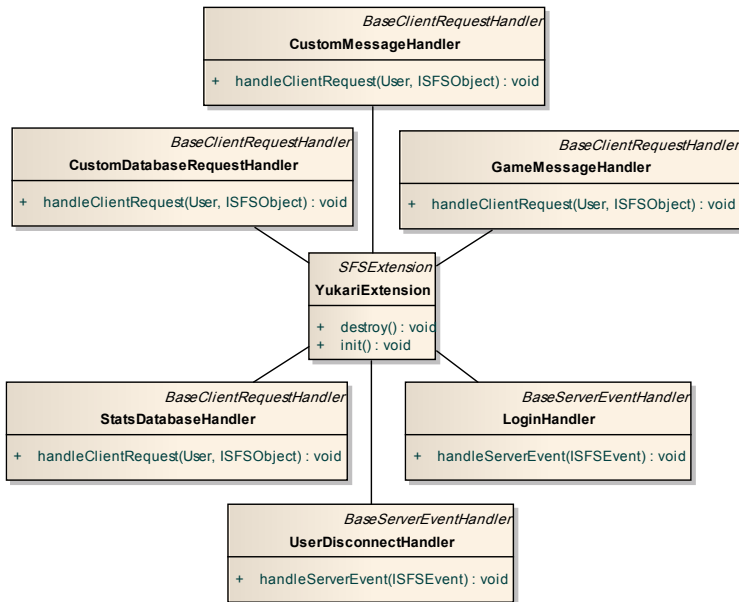
Algoritma pendaftaran secara umum ditunjukkan oleh *flowchart* pada Gambar 3.16. Sebelum melakukan log masuk pada aplikasi permainan, seluruh pengguna harus memastikan dirinya telah terdaftar dalam basis data. Apabila belum terdaftar, maka pengguna tersebut harus melakukan proses pendaftaran dalam aplikasi berbasis web. Secara umum, pengguna harus menuliskan nama akun dan kata kunci yang didaftarkan. Di samping itu, pengguna juga harus menuliskan pertanyaan dan jawaban khusus sebagai kata kunci apabila pengguna lupa kata kuncinya. Untuk memastikan pengguna adalah manusia, pengguna juga harus mencocokkan *captcha*.

3.3.3.2 Algoritma Permainan

Algoritma permainan secara umum ditunjukkan oleh *flowchart* pada Gambar 3.17. Pemain melakukan proses memilih dan mencocokkan nama benda dengan benda selama waktu bermain belum habis, belum ada pemain yang menang, dan pemain tidak keluar dari permainan dengan menekan tombol keluar. Apabila salah satu kondisi yang telah disebutkan terjadi, maka proses memilih dan mencocokkan benda tidak lagi dapat dilakukan.

3.3.3.3 Algoritma Pencocokan Benda

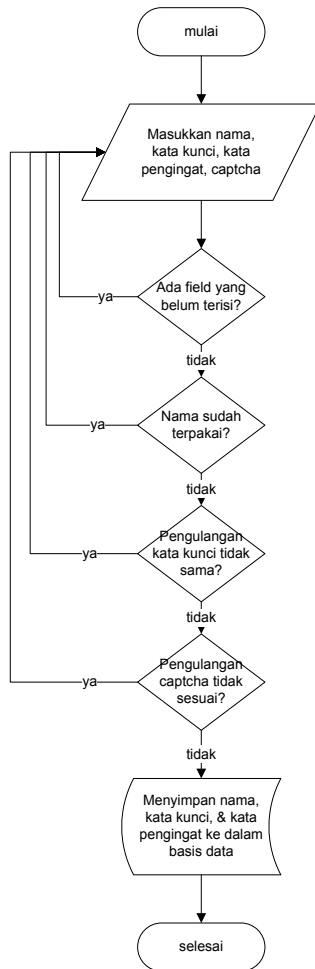
Algoritma pencocokan ditunjukkan oleh *flowchart* pada Gambar 3.18. Pemain melakukan proses memilih dan mencocokkan nama benda dengan benda. Apabila pencocokan berhasil, akan didapatkan penambahan skor. Apabila pencocokan gagal, skor akan berkurang.



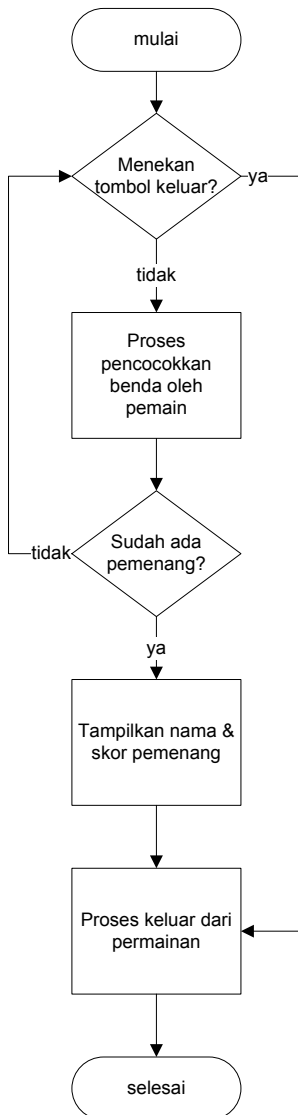
Gambar 3.15 Diagram Hubungan Antarkelas Ekstensi SFS2X

3.3.3.4 Algoritma Pemeriksaan Pemenang

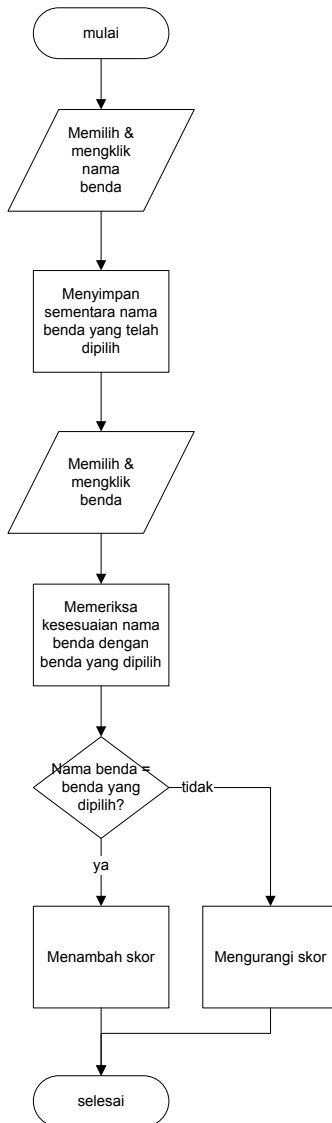
Selama benda dan nama benda yang tersedia pada suatu peta permainan masih belum dicocokkan semua dan waktu bermain juga belum habis, maka pemeriksaan pemain yang keluar sebagai pemenang belum bisa dilakukan. Pemeriksaan pemenang baru dilakukan apabila salah satu kondisi yang telah disebutkan terjadi. Lebih lanjut, algoritma tersebut ditunjukkan dalam Gambar 3.19.



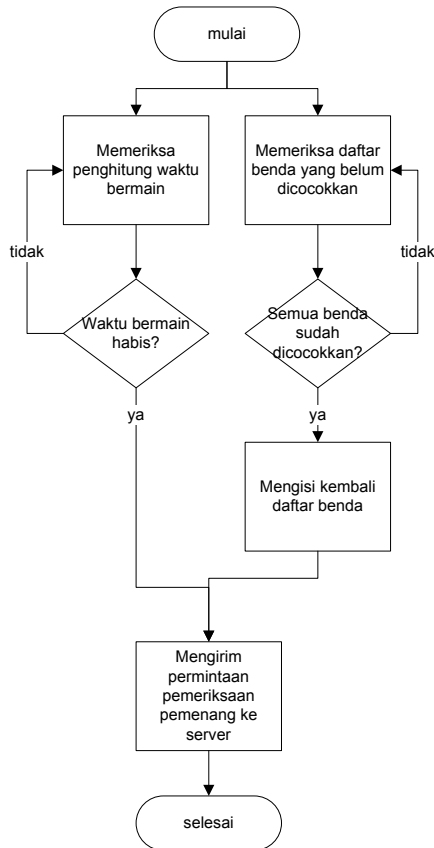
Gambar 3.16 *Flowchart* Pendaftaran Pengguna



Gambar 3.17 *Flowchart Permainan Secara Umum*



Gambar 3.18 *Flowchart* Pencocokan Benda

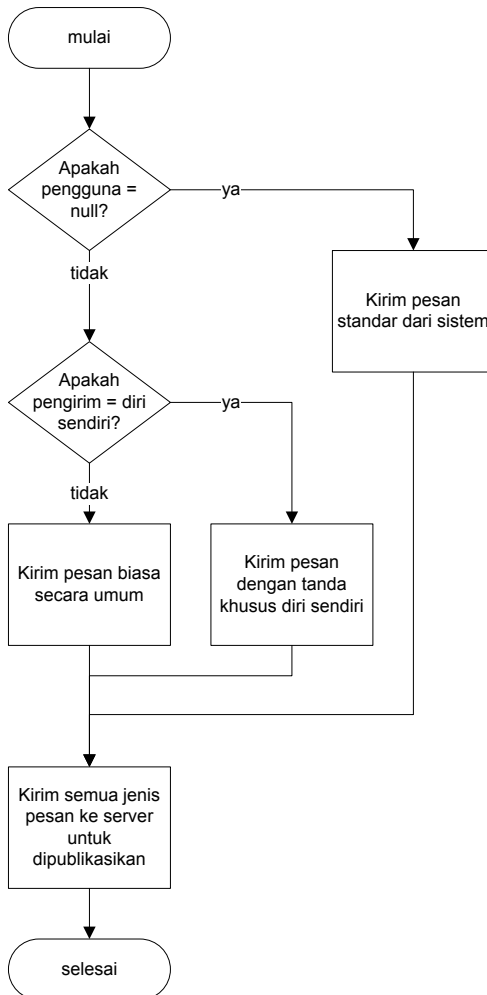


Gambar 3.19 Flowchart Pemeriksaan Pemenang

3.3.3.5 Algoritma Perbincangan Antar Pemain

Perbincangan antar pemain dilakukan melalui kotak perbincangan yang telah disediakan dalam aplikasi. Harus dipastikan terlebih dahulu perbincangan benar-benar muncul dari pemain, bukan dari galat yang terjadi dalam sistem. Kemudian harus dikenali siapa yang menuliskan dan mengirimkan perbincangan, sehingga pada tampilan kepada pengguna juga

dapat dibedakan antara kata-katanya sendiri dengan kata-kata pengguna lain. Lebih lanjut dapat dilihat pada Gambar 3.20.



Gambar 3.20 *Flowchart* Perbincangan Antar Pemain

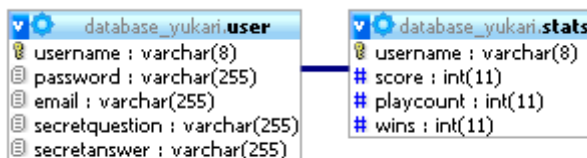
3.3.4 Perancangan Manajemen Data Pengguna

Pada perancangan manajemen data pengguna akan dirancang mengenai struktur basis data pengguna, halaman registrasi, dan halaman memperbarui kata kunci.

3.3.4.1 Perancangan Struktur Basis Data Pengguna

Pada permainan dengan banyak pemain, para pemain akan saling berinteraksi satu sama lain pada lingkungan permainan yang sama dan dalam waktu yang bersamaan. Diperlukan suatu cara agar sistem dapat mengenali pemain satu dan pemain lainnya. Oleh karena itu, setiap pemain harus mempunyai sebuah tanda pengenal unik. Sebelum memainkan permainan, pengguna harus melakukan log masuk terlebih dahulu. Untuk menyimpan data-data pengguna tersebut digunakan basis data MySQL.

Permainan “Yuk Kita Cari” ini juga mempunyai fitur pencatatan statistik permainan penggunaannya. Statistik tersebut digunakan ketika pemain ingin melihat statistik permainan yang telah dilakukannya. Statistik tersebut juga digunakan untuk mendapatkan pemain dengan skor terbanyak yang akan ditampilkan dalam permainan agar terdapat persaingan antar pemain. Gambar 3.21 menunjukkan struktur basis data yang digunakan untuk mengakomodasi kedua kebutuhan tersebut. Tabel *user* berisi tentang informasi yang dibutuhkan untuk log masuk dan memperbarui kata kunci. Sedangkan tabel *stats* berisi variabel-variabel yang digunakan untuk menyimpan statistik permainan pemain.



Gambar 3.21 Struktur Basis Data Permainan “Yuk Kita Cari”

3.3.5 Perancangan Tampilan Antarmuka

Pada perancangan tampilan antarmuka akan dibahas mengenai entitas permainan, peta permainan, dan beberapa rancangan jendela permainan.

3.3.5.1 Perancangan Entitas Permainan

Terdapat banyak benda yang bisa dicocokkan oleh para pemain pada setiap peta permainan. Daftar benda dan kosakata bahasa Inggris untuk setiap benda tersebut terdapat pada Tabel 3.8.

Tabel 3.8 Daftar Benda dan Kosakata Bahasa Inggris

Nama	Kosakata Bahasa Inggris	Peta Permainan
Selang	<i>hose</i>	Teras
Tempat sampah	<i>trashbin</i>	Teras
Sapu	<i>broom</i>	Teras
Bunga	<i>flower</i>	Teras
Pintu	<i>door</i>	Teras
Dinding	<i>wall</i>	Teras
Bola	<i>ball</i>	Ruang Keluarga
Lukisan	<i>picture</i>	Ruang Keluarga
Buku-buku	<i>books</i>	Ruang Keluarga
Lampu	<i>lamp</i>	Ruang Keluarga
Kursi	<i>chair</i>	Ruang Keluarga
Meja	<i>table</i>	Ruang Keluarga
Teko	<i>teapot</i>	Ruang Makan
Garpu	<i>fork</i>	Ruang Makan

Nama	Kosakata Bahasa Inggris	Peta Permainan
Sendok	<i>spoon</i>	Ruang Makan
Cangkir	<i>cup</i>	Ruang Makan
Kulkas	<i>refrigerator</i>	Ruang Makan
Mangkuk	<i>bowl</i>	Ruang Makan
Tempat tidur	<i>bed</i>	Kamar Tidur
Kalender	<i>calendar</i>	Kamar Tidur
Buku-buku	<i>books</i>	Kamar Tidur
Bangku/meja belajar	<i>desk</i>	Kamar Tidur
Selimut	<i>blanket</i>	Kamar Tidur
Jam	<i>clock</i>	Kamar Tidur

3.3.5.2 Perancangan Peta Permainan

Pada permainan “Yuk Kita Cari” terdapat empat macam peta yang dapat dimainkan secara bersama-sama oleh para pemain. Keempat macam peta permainan tersebut adalah teras, ruang keluarga, ruang makan, dan kamar tidur. Keempat peta permainan tersebut masing-masing mempunyai batasan maksimum jumlah pemain dan daftar benda yang berbeda pula. Pada Gambar 3.22 digambarkan secara umum masing-masing rancangan peta permainan.



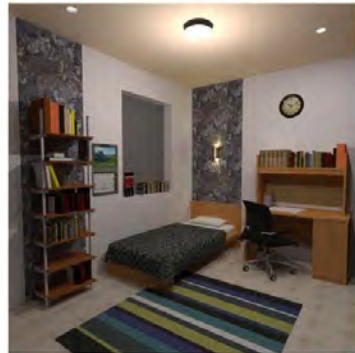
teras



ruang keluarga



ruang makan



kamar tidur

Gambar 3.22 Rancangan Tampilan Peta Permainan

Dari keempat peta permainan tersebut, tingkat kesulitan yang diberikan berbeda-beda. Peta teras dirancang untuk pemula yang baru memainkan permainan “Yuk Kita Cari”. Peta ini dapat dimainkan maksimal oleh dua pemain. Sedangkan peta ruang keluarga dapat dimainkan oleh tiga orang pemain sehingga memungkinkan persaingan yang lebih ketat.

Dua peta berikutnya yaitu ruang makan dan kamar tidur adalah peta yang dapat dimainkan maksimal oleh empat orang pemain. Kedua peta ini mempunyai tingkat persaingan yang hampir sama.

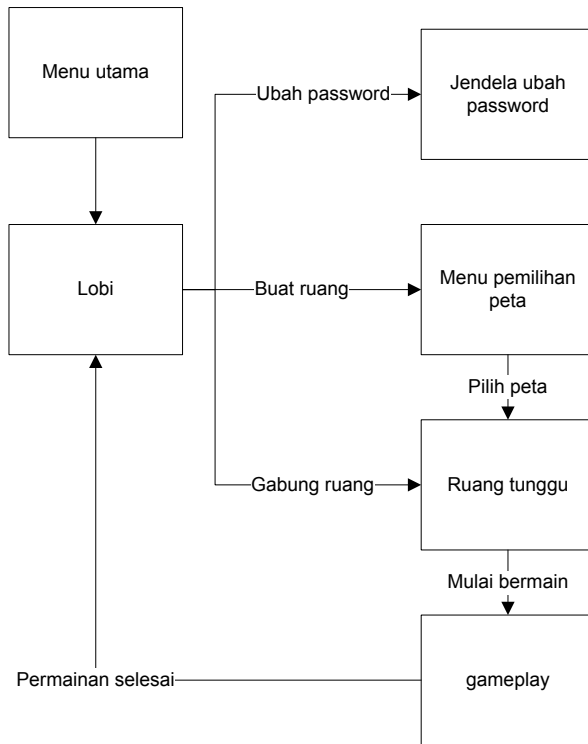
Pada tugas akhir ini ruang permainan dibatasi hanya bisa menampung 4 pemain. Hal ini didasarkan pada keumuman beberapa permainan yang telah ada sebelumnya. Permainan-permainan tersebut mempunyai batasan 4 pemain dalam sebuah ruang permainan. Beberapa daftar permainan yang mempunyai batasan maksimal empat pemain dalam satu ruang permainan terdapat pada Tabel 3.9.

Tabel 3.9 Contoh Permainan Dengan Batasan Maksimal 4 Pemain

No	Nama Permainan	Batasan
1	Zynga texas hold em poker	4 pemain
2	Solar chief	2-4 pemain
3	Mahjong dynasty	4 pemain

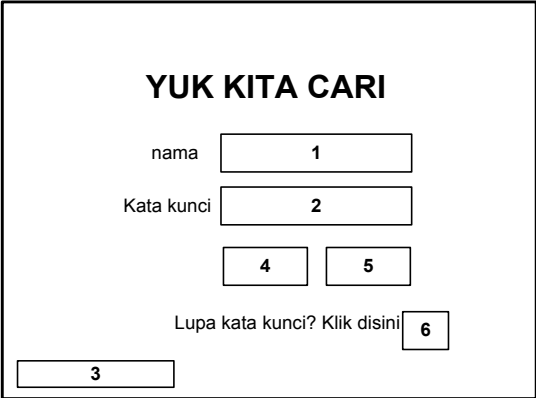
3.3.5.3 Perancangan Jendela Permainan

Dalam permainan “Yuk Kita Cari”, terdapat beberapa macam jendela. Jendela-jendela tersebut adalah menu utama, lobi, jendela ubah kata kunci, jendela pemilihan map, ruang tunggu, dan jendela permainan. Alur perpindahan antar jendela beserta kondisinya terdapat pada Gambar 3.23. Adapun perancangan bentuk dan bagian masing-masing jendela serta penjelasan detailnya terdapat pada Tabel 3.10. Dalam jendela permainan terdapat gambar karakter kucing berwarna kuning yang diunduh dan boleh digunakan secara gratis (tidak berbayar) dari alamat situs <http://yurikel11.deviantart.com/art/Cute-Animals-Vector-135409537>.

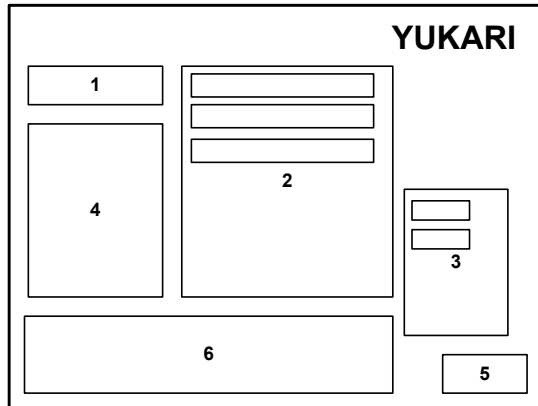


Gambar 3.23 Alur Perpindahan Antar Jendela

Tabel 3.10 Keterangan Rancangan Antarmuka

Nama Jendela	Gambar dan Keterangan
Menu Utama	<p data-bbox="370 296 501 323">Deskripsi:</p> <p data-bbox="370 328 964 560">Menu utama adalah tampilan awal ketika pengguna membuka aplikasi. Menu utama digunakan untuk melakukan log masuk ke dalam aplikasi dan juga berisi tautan untuk melakukan pendaftaran dan memperbarui kata kunci. Rancangan menu utama dapat dilihat pada Gambar 3.24.</p> <div data-bbox="398 600 934 999"><p>The screenshot shows a login and registration interface. At the top, the text 'YUK KITA CARI' is displayed. Below it, there are two input fields: one labeled 'nama' with a small box labeled '1' next to it, and another labeled 'Kata kunci' with a small box labeled '2' next to it. Below these fields are two buttons: one labeled '4' and another labeled '5'. Below the buttons is a link that says 'Lupa kata kunci? Klik disini' with a small box labeled '6' next to it. At the bottom left, there is a small box labeled '3'.</p></div> <p data-bbox="398 1007 934 1034">Gambar 3.24 Rancangan Jendela Menu Utama</p> <p data-bbox="370 1074 529 1101">Keterangan:</p> <ol data-bbox="370 1106 734 1305" style="list-style-type: none">1. <i>Field</i> nama pemain.2. <i>Field</i> kata kunci.3. Info <i>error</i> ketika log masuk.4. Tombol log masuk.5. Tombol pendaftaran.6. Tombol lupa kata kunci.
"Lobby"	Deskripsi:

Lobby adalah tampilan ketika pengguna telah melakukan log masuk. Jendela tersebut merupakan jendela utama untuk menuju jendela-jendela lainnya. Rancangan jendela “*Lobby*” dapat dilihat pada Gambar 3.25.



Gambar 3.25 Rancangan Jendela “*Lobby*”

Keterangan:

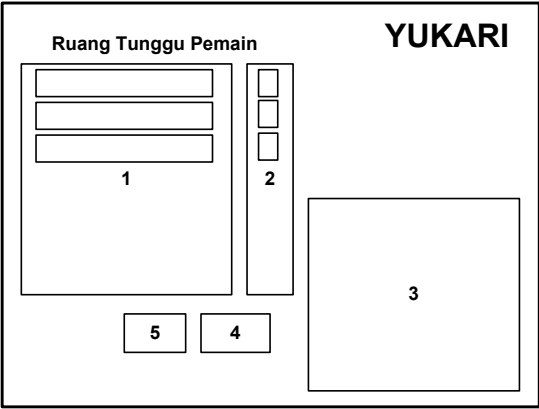
1. Nama pemain.
2. Daftar ruangan permainan yang telah dibuat oleh pemain lain.
3. Jumlah dan daftar pemain yang berada di “*Lobby*”.
4. Tombol-tombol panel yang digunakan untuk menuju jendela lain.
5. Tombol *logout*.
6. Kotak perbincangan.

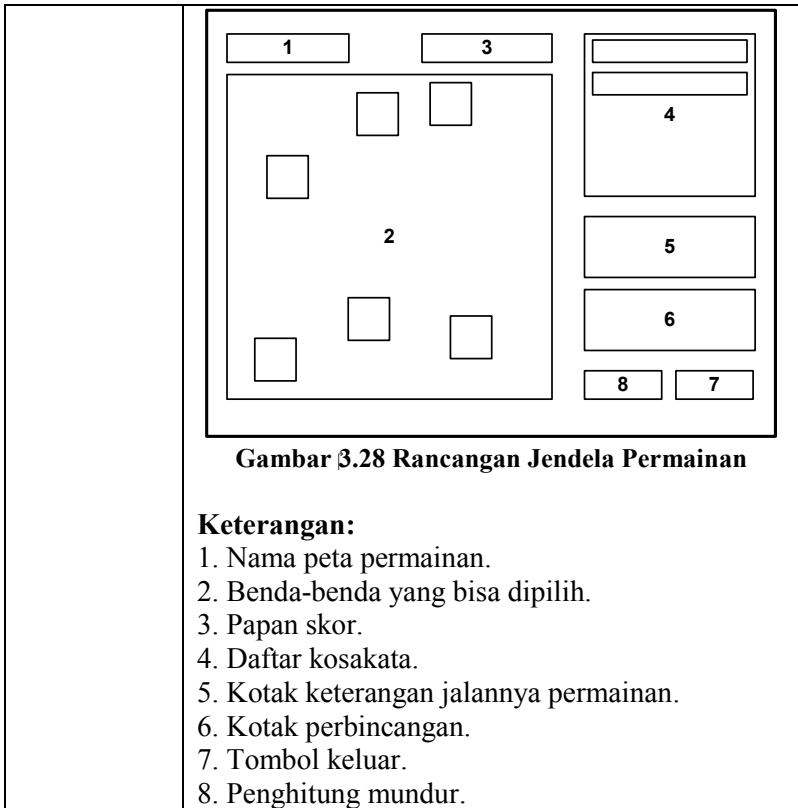
Pemilihan Peta/Lokasi Permainan

Deskripsi:

Jendela pemilihan peta adalah jendela untuk memilih peta mana yang diinginkan ketika membuat ruangan permainan. Rancangan jendela pemilihan peta dapat dilihat pada Gambar 3.26.

	<div><div><div><div><div>YUKARI</div><div>Menu Pilih Peta</div><div><div><div></div><div></div><div></div></div><div>1</div></div><div><div>3</div><div>2</div></div></div></div></div></div>
<div>Ruang Tunggu</div>	<div><div>Gambar 3.26 Rancangan Jendela Pemilihan Peta</div><div>Keterangan:<div><div>1. Daftar peta permainan yang dapat dipilih.</div><div>2. Tombol <i>Create Room</i>.</div><div>3. Tombol <i>Cancel</i>.</div></div></div><div><div>Deskripsi:</div><div>Ruang tunggu adalah jendela tempat pemain pembuat ruang permainan menunggu pemain lain untuk bergabung ke dalam ruang permainan yang telah dibuat. Dalam ruangan ini juga dilengkapi dengan kotak perbincangan. Rancangan jendela ruang tunggu dapat dilihat pada Gambar 3.27.</div></div></div>

	 <p>Gambar 3.27 Rancangan Jendela Ruang Tunggu</p> <p>Keterangan:</p> <ol style="list-style-type: none"> 1. Daftar pemain dalam ruangan. 2. Warna aliansi. 3. Kotak perbincangan. 4. Tombol <i>Start Game</i>. 5. Tombol <i>Cancel</i>.
Permainan	<p>Deskripsi:</p> <p>Jendela permainan adalah jendela untuk memainkan permainan. Rancangan jendela permainan dapat dilihat pada Gambar 3.28.</p>



BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak yang didasarkan pada Bab III.

4.1. Lingkungan Implementasi

Lingkungan pembangunan untuk aplikasi ini menggunakan komputer tunggal. Berikut adalah spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam lingkungan pembangunan.

Spesifikasi perangkat keras:

- Processor Intel Core 2 Duo T6500 @ 2,10 GHz
- Memory 2012 MB
- Kartu Grafis Mobile Intel 4 Series Express Chipset Family

Spesifikasi perangkat lunak:

- Windows XP Professional SP 2
- Adobe Flash CS5
- Flash Develop 4.0
- SmartFoxServer2X Versi 2.0.1
- Netbeans 6.9
- Java Development Kit 1.7.0
- Xampp 1.7.4, yang berisi:
 - Apache 2.2.17
 - MySQL 5.5.8
 - PHP 5.3.5.

4.2. Implementasi Antarmuka

Dalam subbab ini akan dibahas tentang implementasi antarmuka dari analisis dan perancangan perangkat lunak. Implementasi tersebut akan ditampilkan dalam bentuk potongan gambar antarmuka.

4.2.1. Jendela Menu Utama

Seperti yang telah dijelaskan pada Tabel 3.10, jendela menu utama adalah antarmuka yang digunakan untuk melakukan log masuk kedalam aplikasi. Jendela tersebut juga berisi tautan untuk melakukan pendaftaran dan memperbarui kata kunci. Implementasi antarmuka jendela menu utama digambarkan pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka Menu Utama

4.2.2. Jendela “Lobby”

Seperti yang telah dijelaskan pada Tabel 3.10, jendela “Lobby” adalah antarmuka ketika pengguna telah melakukan log

masuk. Berguna sebagai pintu utama menuju jendela-jendela lainnya. Pada jendela ini pemain dapat melakukan perbincangan, membuat ruang permainan, bergabung ke dalam ruang permainan, melihat statistik dan mengganti kata kunci. Implementasi antarmuka jendela “*Lobby*” dapat dilihat pada Gambar 4.2.

4.2.3. Jendela Pemilihan Peta Permainan

Jendela pemilihan peta permainan adalah antarmuka dimana pemain memilih peta yang ingin dimainkan ketika membuat ruang permainan. Ruang permainan dibuat setelah pemain memilih peta. Gambar 4.3 menunjukkan implementasi jendela pemilihan peta permainan.

4.2.4. Jendela Ruang Tunggu

Jendela ruang tunggu adalah antarmuka dimana pemain pembuat ruang permainan menunggu pemain lain untuk bergabung ke dalam ruang permainan yang telah dibuat. Dalam ruang ini juga dilengkapi dengan kotak perbincangan. Gambar 4.4 menunjukkan implementasi jendela ruang tunggu.

4.2.5. Jendela Permainan

Seperti yang telah dijelaskan pada Tabel 3.10, jendela permainan adalah jendela untuk memainkan permainan “Yuk Kita Cari” bersama pengguna yang lain. Implementasi antarmuka jendela permainan digambarkan pada Gambar 4.5.

4.2.6. Implementasi Pendaftaran

Implementasi sistem pendaftaran akan diawali dengan membuat basis data, kemudian diikuti dengan membuat sebuah halaman web untuk pendaftaran pengguna. Halaman web pendaftaran dibuat menggunakan PHP. Gambar 4.6 menunjukkan halaman web pendaftaran.

Pada halaman pendaftaran, proses logika yang dilakukan adalah melakukan pemeriksaan terhadap masukan pengguna sebelum data pengguna dimasukkan ke dalam basis data.

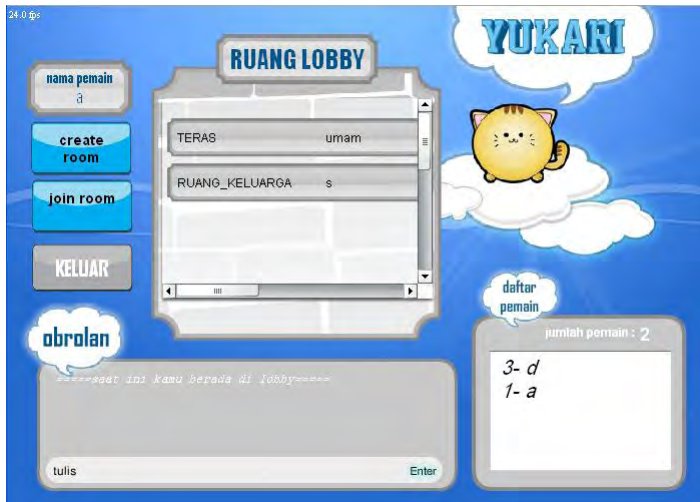
Pemeriksaan terdiri dari dua tahap. Tahap awal dilakukan dengan Javascript, untuk memeriksa apakah ada ruas yang kosong, panjang nama pengguna (4-8 karakter alfabet), panjang kata kunci (6-20 karakter), dan format surat elektronik.

Tahap pemeriksaan kedua dilakukan setelah pengguna menekan tombol daftar. Pemeriksaan kedua melibatkan basis data. Yang pertama-tama memeriksa terlebih dahulu apakah *captcha* yang dimasukkan telah benar. Selanjutnya memeriksa apakah nama pengguna yang dimasukkan telah ada pada basis data.

4.2.7. Implementasi Pembaruan Kata Kunci

Implementasi fitur pembaruan kata kunci dilakukan dengan menggunakan halaman web. Tampilan dari halaman lupa kata kunci dapat dilihat pada Gambar 4.6 Tampilan Halaman Pendaftaran & Halaman Lupa Kata Kunci.

Proses logika pada implementasi fitur pembaruan kata kunci adalah pemeriksaan masukan pengguna sebelum melakukan perubahan pada kata kunci pengguna. Pemeriksaan dilakukan setelah pengguna menekan tombol daftar. Pertama-tama, sistem memeriksa terlebih dahulu apakah *captcha* yang dimasukkan telah benar. Selanjutnya adalah memeriksa apakah nama pengguna yang dimasukkan telah ada pada basis data. Pemeriksaan yang ketiga adalah memastikan apakah pertanyaan dan jawaban rahasia yang dimasukkan oleh pengguna sama dengan yang ada pada basis data.



Gambar 4.2 Implementasi Jendela “Lobby”



Gambar 4.3 Implementasi Jendela Pemilihan Peta Permainan



Gambar 4.4 Implementasi Jendela Ruang Tunggu



Gambar 4.5 Implementasi Jendela Permainan

Register for Yukari Multiplayer	Lupa Kata Kunci Permainan Yuk Kita Cari
Username (4-8 character alphabet only) <input type="text" value="fakhr0"/>	Username <input type="text"/>
Password (6-20 character) <input type="password" value="*****"/>	E-mail <input type="text"/>
Repeat Password <input type="password" value="*****"/>	Secret Question <input type="text" value="What is the name of your best friend from childhood?"/>
E-mail <input type="text" value="umamlearn@gmail.com"/>	Secret Answer <input type="text" value="oilk"/>
Secret Question <input type="text" value="What is the name of your best friend from childhood?"/>	Input text below: <div> <div>52XR8</div> <div>52x8</div> </div> <input type="text"/>
<input type="button" value="Register"/>	<input type="button" value="Request New Password"/>

Gambar 4.6 Tampilan Halaman Pendaftaran & Halaman Lupa Kata Kunci

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dibahas pengujian dan evaluasi dari perangkat lunak dan implementasi permainan “Yuk Kita Cari”.

5.1 Lingkungan Pelaksanaan Pengujian

Lingkungan pengujian merupakan komputer tempat pengujian sistem dilakukan. Aplikasi diakses oleh 50 pemain. Pengujian dilakukan secara hampir bersamaan. Aplikasi ini diuji dalam lingkungan pengujian dengan spesifikasi sebagai berikut:

Prosesor	: Intel Core 2 Duo T6500 @ 2,10 GHz
Memori	: 2012 MB
Kartu Grafis	: Mobile Intel(R) 4 Series Express Chipset Family
Sistem Operasi	: Microsoft Windows XP Professional SP2
Jaringan	: Jaringan lokal

5.2 Dasar Pengujian

Pengujian pertama pada permainan “Yuk Kita Cari” menggunakan metode pengujian *black box* yang berfokus pada kebutuhan fungsional perangkat lunak tersebut. Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar telah diimplementasikan dan bekerja sebagaimana mestinya. Pengujian yang kedua adalah pengujian performa untuk mengetahui performa server dan aplikasi klien ketika digunakan banyak pemain.

5.3 Skenario Pengujian

Pada subbab ini akan dijelaskan mengenai skenario pengujian dari permainan berbasis web “Yuk Kita Cari”.

5.3.1 Pengujian Fungsional

Berikut ini akan dijelaskan mengenai pengujian yang dilakukan pada permainan berbasis web “Yuk Kita Cari” berdasarkan kebutuhan fungsional yang telah dijabarkan pada Bab III.

5.3.1.1 Pengujian Menentukan Lokasi Permainan

Pada bagian ini akan dijelaskan proses pengujian untuk menentukan lokasi permainan sebelum bermain. Tabel 5.1 menyajikan prosedur pengujian proses menentukan lokasi permainan.

Tabel 5.1 Prosedur Pengujian Proses Menentukan Lokasi Permainan

Test ID		UCT001		
Tujuan Test		Mengecek apakah fungsi menentukan lokasi permainan berjalan normal.		
Kondisi Awal		Pemain pembuat sudah berhasil melakukan log masuk ke dalam antarmuka “Lobby”.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Menekan tombol “Create Room”, memilih lokasi permainan	Memasukkan data input.	Ruang baru terbentuk sesuai dengan lokasi yang telah dipilih pemain pembuat,	Sistem menampilkan hasil yang diharapkan.	Proses menentukan lokasi permainan berhasil.

“Teras” dan menekan tombol “Create-Room” pada jendela pilihan lokasi.		pemain pembuat berpindah ke antarmuka “Ruang Tunggu”, pemain yang lain melihat ruang baru pada antarmuka “Lobby”.		
---	--	---	--	--

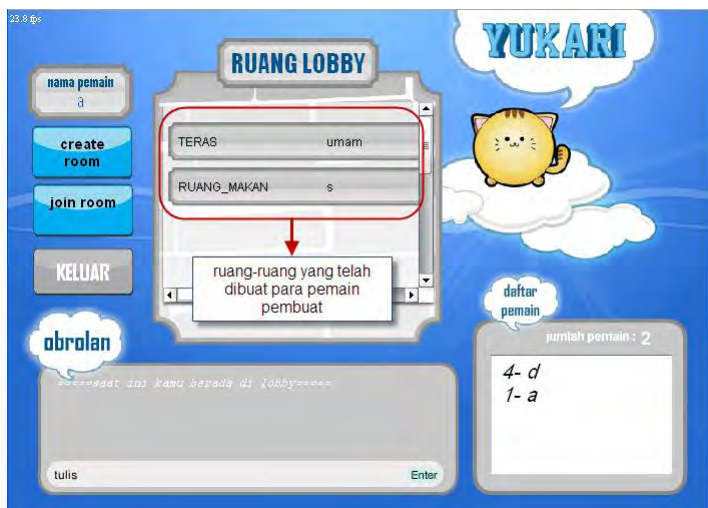
Proses pengujian ini dikatakan berhasil jika pemain pembuat berhasil pindah ke antarmuka ruang tunggu dan pemain lain telah melihat ruang baru pada antarmuka “Lobby” sebagaimana yang ditunjukkan pada Gambar 5.1 dan Gambar 5.2.



Gambar 5.1 Tampilan Pemain Pembuat Jika Berhasil Menentukan Lokasi Permainan

5.3.1.2 Pengujian Bergabung ke Dalam Ruang

Pada bagian ini akan dijelaskan proses pengujian untuk memilih ruang yang telah tersedia dan bergabung ke dalam ruang tersebut. Tabel 5.2 menyajikan prosedur pengujian proses bergabung ke dalam ruang.



Gambar 5.2 Tampilan Pemain Lain Jika Pemain Pembuat Telah Menentukan Lokasi Permainan

Tabel 5.2 Prosedur Pengujian Proses Bergabung ke Dalam Ruang

Test ID	UCT002
Tujuan Test	Mengecek apakah fungsi bergabung ke dalam ruang berjalan normal.
Kondisi Awal	1. Pemain biasa sudah berhasil melakukan log masuk ke dalam “Lobby”. 2. Ada ruang yang telah dibuat oleh pemain pembuat. 3. Ruang-ruang yang telah ada belum penuh.

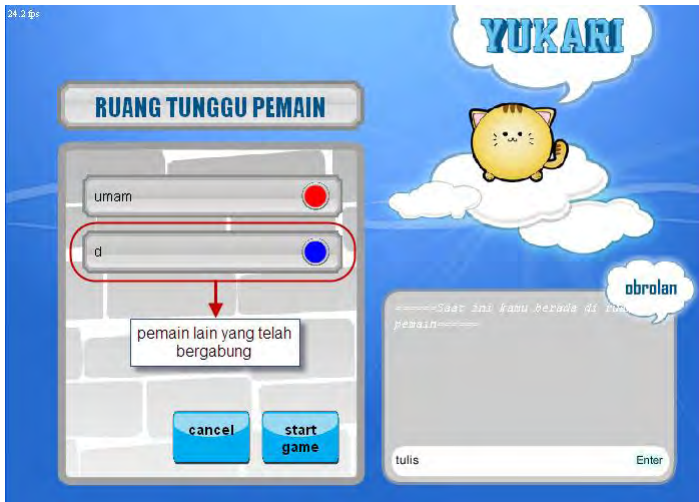
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Memilih ruangan “Teras” yang tersedia dalam daftar, lalu menekan tombol “Join Room” pada jendela “Lobby”.	Memasukkan data input.	Pemain biasa berpindah ke antarmuka “Ruang Tunggu”, pemain pembuat melihat ada pemain baru pada antarmuka ruang tunggu.	Sistem menampilkan hasil yang diharapkan.	Proses bergabung ke dalam ruang berhasil.

Proses pengujian ini dikatakan berhasil jika pemain biasa berhasil pindah ke antarmuka ruang tunggu dan pemain pembuat telah melihat pemain baru pada antarmuka ruang tunggu sebagaimana yang ditunjukkan pada Gambar 5.3 dan Gambar 5.4.

5.3.1.3 Pengujian Memulai Permainan

Pada bagian ini akan dijelaskan proses pengujian untuk memulai permainan oleh pemain pembuat. Tabel 5.3 menyajikan prosedur pengujian proses memulai permainan.

Proses pengujian ini dikatakan berhasil jika semua pemain berhasil pindah ke antarmuka permainan sebagaimana yang ditunjukkan pada Gambar 5.5.



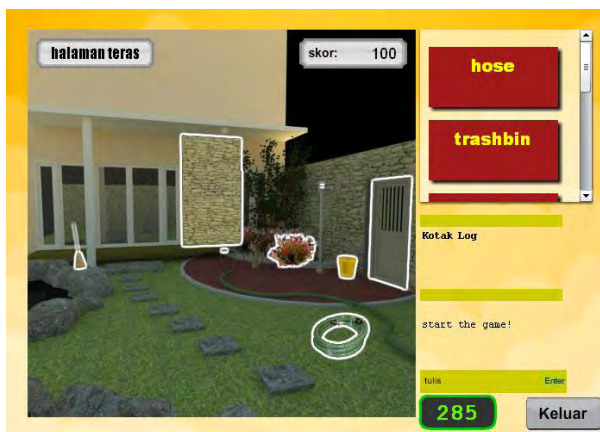
Gambar 5.3 Tampilan Ruang Tunggu pada Pemain Pembuat



Gambar 5.4 Tampilan Ruang Tunggu pada Pemain Biasa

Tabel 5.3 Prosedur Pengujian Proses Memulai Permainan

Test ID		UCT003		
Tujuan Test		Mengecek apakah fungsi memulai permainan berjalan normal.		
Kondisi Awal		1. Pemain pembuat sudah berhasil membuat ruang dan masuk ke “Ruang Tunggu”. 2. Ada minimal 2 pemain dalam “Ruang Tunggu”.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Pemain pembuat menekan tombol “ <i>Start Game</i> ” pada antarmuka “Ruang Tunggu”.	Memasukkan data input.	Pemain pembuat dan pemain biasa berpindah dari antarmuka “Ruang Tunggu” ke antarmuka permainan.	Sistem menampilkan hasil yang diharapkan.	Proses memulai permainan berhasil.

**Gambar 5.5 Tampilan Antarmuka Permainan pada Semua Pemain**

5.3.1.4 Pengujian Fitur Perbincangan

Pada bagian ini akan dijelaskan proses pengujian fitur perbincangan oleh semua pemain. Pengujian dilakukan untuk mengecek apakah fungsi perbincangan berjalan normal. Prosedur pengujian proses memulai permainan dapat dilihat pada Tabel 5.4.

Tabel 5.4 Prosedur Pengujian Proses Memulai Permainan

Test ID		UCT004		
Tujuan Test		Mengecek apakah fungsi perbincangan berjalan normal.		
Kondisi Awal		Pemain pembuat atau pun biasa sudah berhasil melakukan log masuk ke dalam “Lobby”.		
Data Input	Prose- dur peng- ujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesim- pulan
Pemain “s”menetikkan kata-kata “assalamu-alaikum” dalam kotak perbincangan kemudian menekan tombol “Enter”. Kemudian pemain “a” menjawab dengan mengetikkan “waalaikum	Me- masuk- kan data input.	Kata-kata atau kalimat yang diketikkan pemain terkirim ke pemain-pemain lain, semua pemain di “Lobby” dapat membaca kata atau kalimat yang dikirim pemain lain sehingga	Sistem menampilkan hasil yang diharapkan.	Proses perbin- cangan berhasil.

ussalam” dan menekan tombol “Enter”.		seluruh pemain dapat berbincang- bincang.		
--	--	--	--	--

Berikut ini adalah potongan gambar pengujian yang dijalankan. Proses pengujian ini dikatakan berhasil jika semua pemain dapat saling berbincang dan melihat perkataan pemain lain sebagaimana yang ditunjukkan pada Gambar 5.6.



Gambar 5.6 Tampilan Perbincangan Antar Pemain

5.3.1.5 Pengujian Mencocokkan Benda

Pada bagian ini akan dijelaskan proses mencocokkan benda selama permainan oleh semua pemain. Tabel 5.5 menyajikan prosedur pengujian proses mencocokkan benda.

Tabel 5.5 Prosedur Pengujian Proses Mencocokkan Benda

Test ID	UCT005
Tujuan Test	Mengecek apakah fungsi mencocokkan

		benda berjalan normal.		
Kondisi Awal		Semua pemain sudah masuk ke dalam antarmuka permainan.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Pemain “a” memilih nama benda “Door” kemudian mengklik gambar pintu.	Memasukkan data input.	benda yang sudah dicocokkan ditandai dengan tampilan nama pemain yang sudah mencocokkan, skor pemain tersebut bertambah.	Sistem menampilkan hasil yang diharapkan.	Proses mencocokkan benda berhasil.

Berikut ini adalah potongan gambar pengujian yang dijalankan. Proses pengujian ini dikatakan berhasil jika benda yang sudah dicocokkan tidak bisa diklik lagi dan telah ditandai dengan nama pemain yang mencocokkan. Selain itu, skor pemain yang mencocokkan bertambah, sebagaimana yang ditunjukkan pada Gambar 5.7.



Gambar 5.7 Tampilan Setelah Mencocokkan Benda

5.3.2 Pengujian Performa

Pengujian ini bertujuan untuk mengamati performa aplikasi ketika dijalankan dan menguji kemampuan server dalam menangani banyak pemain. Pengujian ini dilakukan dengan menjalankan aplikasi klien dalam jumlah yang banyak dan memantau panel SmartFoxServer2X.

5.3.2.1 Pengujian Performa Klien (UCT006)

Pengujian dilakukan dengan menjalankan aplikasi klien dan melakukan beberapa proses dalam permainan. Gambar 5.8 menunjukkan tampilan klien ketika mengetikkan nama pengguna dan kata kunci untuk melakukan log masuk. Pada bagian pojok kiri atas dapat dilihat performa klien cukup stabil, rata-rata 24 fps (*frame per second*). Demikian pula ketika klien berada pada halaman “Lobby” dan memilih ruangan, performa cenderung tetap yakni rata-rata 24 fps. Gambar 5.9 menunjukkan pengukuran performa aplikasi klien saat masuk halaman “Lobby”, sedangkan Gambar 5.10 menunjukkan pengukuran performa aplikasi klien saat memilih ruangan.

Sedangkan pengujian berikutnya dilakukan pada saat klien berada di ruang tunggu untuk menunggu pemain yang lain bergabung dan saat permainan dimulai. Saat klien berada di ruang tunggu, performa klien cenderung tetap yakni 24 fps. Salah satu sebabnya adalah pembatasan jumlah klien yang bergabung dalam suatu ruang maksimal sebanyak 4 pemain. Dengan adanya pembatasan tersebut, meski klien yang terhubung ke server sangat banyak, namun klien-klien tadi tidak akan bergabung dalam satu ruang sekaligus secara keseluruhan. Gambar 5.11 menunjukkan pengukuran performa aplikasi klien pada halaman ruang tunggu. Efek pembatasan jumlah pemain dalam satu ruang juga berdampak pada performa aplikasi klien saat permainan dimulai. Performa klien tidak mengalami perubahan yang berarti, yakni tetap rata-rata 24 fps.

Gambar 5.12 menunjukkan pengukuran performa aplikasi klien pada saat permainan dimulai.



Gambar 5.8 Tampilan Pengukuran Performa Klien pada Halaman Log Masuk



Gambar 5.9 Tampilan Pengukuran Performa Klien pada Halaman Lobby



Gambar 5.10 Tampilan Pengukuran Performa Klien pada Halaman Memilih Lokasi Permainan



Gambar 5.11 Tampilan Pengukuran Performa Klien pada Halaman Ruang Tunggu Pemain

5.3.2.2 Pengujian Performa Server

Pengujian dilakukan dengan menjalankan banyak aplikasi klien dan memantau panel SmartFoxServer2X. Pengujian melibatkan 50 klien yang seluruhnya terhubung bersamaan ke dalam server.



Gambar 5.12 Tampilan Pengukuran Performa Klien pada Halaman Permainan

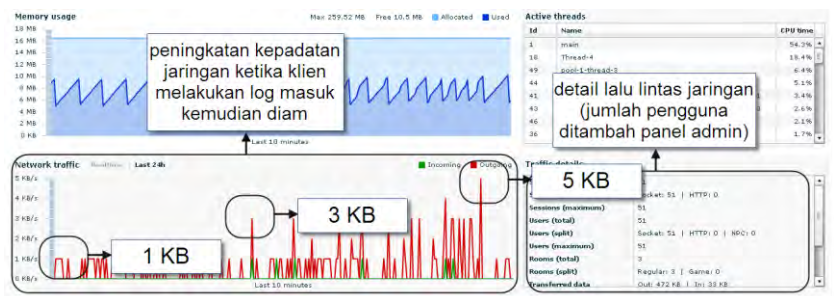
5.3.2.2.1 Pengujian Performa Saat Klien Dalam Kondisi Diam (UCT007)

Pengujian yang pertama dilakukan saat klien-klien baru saja melakukan log masuk kemudian tidak melakukan aktifitas apa pun. Log masuk oleh 50 klien dilakukan secara bertahap. Gambar 5.13 menunjukkan kondisi diam pada halaman “Lobby” setelah semua klien melakukan log masuk. Dalam kondisi tersebut sudah nampak adanya peningkatan kepadatan lalu lintas jaringan dari yang sebelumnya hanya 1 KB/detik menjadi hingga 5 KB/detik. Rata-rata kecepatan transfer data sebesar 0,47 KB/detik. Sementara itu, penggunaan memori komputer oleh SmartFoxServer2X tercatat tidak mengalami

peningkatan, relatif tetap pada kisaran 6 sampai 10 MB. Dengan demikian, kondisi tersebut masih dapat ditangani dengan baik oleh server. Gambar 5.14 menunjukkan grafik peningkatan kepadatan lalu lintas jaringan pada panel SmartFoxServer2X saat 50 klien telah melakukan log masuk dan tidak melakukan apa pun pada halaman “Lobby”.



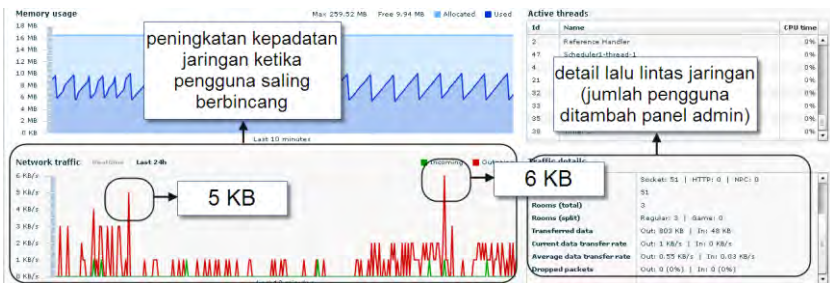
Gambar 5.13 Tampilan Salah Satu Klien Ketika 50 Klien Terhubung ke Dalam Server



Gambar 5.14 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Dalam Kondisi Diam

5.3.2.2 Pengujian Performa Saat Klien Melakukan Perbincangan (UCT008)

Pengujian dilakukan saat seluruh klien menggunakan fitur kotak perbincangan untuk berbincang-bincang dengan klien-klien lain pada halaman “*Lobby*”. Dalam kondisi tersebut nampak ada sedikit peningkatan kepadatan lalu lintas jaringan dari yang sebelumnya pada kondisi diam 5 KB/detik menjadi 6 KB/detik. Rata-rata kecepatan transfer data sebesar 0,55 KB/detik. Sementara itu, penggunaan memori komputer oleh SmartFoxServer2X tercatat tidak mengalami peningkatan, masih relatif tetap pada kisaran 6 sampai 10 MB. Dengan demikian, kondisi seluruh pengguna berbincang-bincang masih dapat ditangani dengan baik oleh server. Gambar 5.15 menunjukkan grafik peningkatan kepadatan lalu lintas jaringan pada panel SmartFoxServer2X saat 50 klien saling berbincang pada halaman “*Lobby*”.



Gambar 5.15 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Saling Berbincang

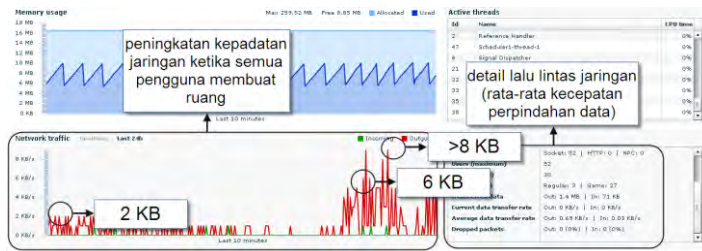
5.3.2.2.3 Pengujian Performa Saat Klien Membuat Ruangan (UCT009)

Pengujian dilakukan saat seluruh klien membuat ruangan permainan, memilih lokasi, kemudian menunggu di ruang tunggu. Gambar 5.16 menunjukkan banyak ruang yang telah dibuat yang terlihat oleh salah seorang klien. Dalam

kondisi tersebut nampak adanya peningkatan signifikan kepadatan lalu lintas jaringan dari yang sebelumnya 5 hingga 6 KB/detik menjadi hingga lebih dari 8 KB/detik. Rata-rata kecepatan transfer data meningkat menjadi sebesar 0,69 KB/detik. Sementara itu, penggunaan memori komputer oleh SmartFoxServer2X tercatat tidak mengalami peningkatan, relatif tetap pada kisaran 6 sampai 10 MB. Dengan demikian, kondisi tersebut juga masih dapat ditangani dengan baik oleh server. Gambar 5.17 menunjukkan grafik peningkatan kepadatan lalu lintas jaringan pada panel SmartFoxServer2X saat 50 klien membuat ruang dan memilih lokasi permainan.



Gambar 5.16 Tampilan Klien ke-51 Ketika 50 Klien Lainnya Membuat Ruang



Gambar 5.17 Halaman Pantau Panel Administrator SmartFoxServer2X dengan 50 Klien Membuat Ruang

5.4 Evaluasi Pengujian

Dari skenario pengujian kebutuhan fungsional dan pengujian performa yang telah dilakukan pada subbab 5.3, dapat diberikan paparan evaluasi sebagai berikut:

1. Proses menentukan lokasi permainan dan membuat ruang telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian (UCT001), dimana detil pengujian dipaparkan secara rinci.
2. Proses untuk bergabung ke dalam ruang bagi pemain biasa telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian (UCT002), dimana detil pengujian dipaparkan secara rinci.
3. Proses untuk memulai permainan bagi pemain pembuat ruang telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian (UCT003), dimana detil pengujian dipaparkan secara rinci.
4. Proses melakukan perbincangan antar pemain telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian (UCT004), dimana detil pengujian dipaparkan secara rinci.
5. Proses mencocokkan benda selama permainan juga telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian (UCT005), dimana detil pengujian dipaparkan secara rinci dalam tabel.
6. Pengujian performa aplikasi klien (UCT006) menunjukkan hasil yang baik dan relatif stabil. Hal tersebut disebabkan oleh pembatasan jumlah pemain dalam satu ruang permainan, yakni maksimal 4 pemain. Meskipun banyak klien yang terhubung, namun pada masing-masing ruang, aplikasi klien hanya menangani 2 hingga 4 pemain saja.
7. Pengujian performa server UCT007 hingga UCT009 juga menunjukkan hasil yang baik. SmartFoxServer2X terbukti mampu mengatur lalu lintas dan sinkronisasi data dalam kondisi 50 klien terhubung dan melakukan berbagai macam

proses. Hal tersebut disebabkan sisi server tidak terlalu banyak menangani proses-proses yang berkaitan dengan aturan permainan. Proses-proses yang berkaitan dengan aturan permainan seperti pengecekan pemenang, pengecekan pencocokan benda, dan lain sebagainya telah ditangani oleh aplikasi klien.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Perancangan aturan permainan telah berhasil dilakukan dan dapat berjalan sesuai dengan hasil pengujian fungsional UCT001 hingga UCT005.
2. Aturan-aturan permainan yang telah dirancang berhasil diimplementasikan dalam sebuah aplikasi klien menggunakan Flash ActionScript 3.0. Integrasi SmartFoxServer2X ke dalam aplikasi klien juga berhasil dilakukan dengan baik.
3. Aplikasi klien terbukti memiliki performa yang baik untuk dimainkan rata-rata 24 fps (*frame per second*) sesuai dengan pengujian performa klien UCT006.
4. SmartFoxServer2X terbukti mampu menangani dengan baik setidaknya 50 klien yang terhubung dengan rata-rata kecepatan transfer data 0,47 KB/detik sampai 0,69 KB/detik sesuai dengan pengujian performa server UCT007 hingga UCT009.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan

pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Pengembangan permainan yang lebih variatif, yaitu dengan penambahan jenis peta atau lokasi permainan dan penambahan daftar benda yang dicocokkan untuk menambah kosa kata bahasa Inggris dan bahasa Arab.
2. Tugas akhir ini menggunakan SmartFoxServer2X Community Edition yang membatasi *concurrent user* sebanyak 100 pengguna. Pengembangan di masa mendatang dapat menggunakan SmartFoxServer Premium Edition agar dapat menangani lebih banyak lagi pengguna.

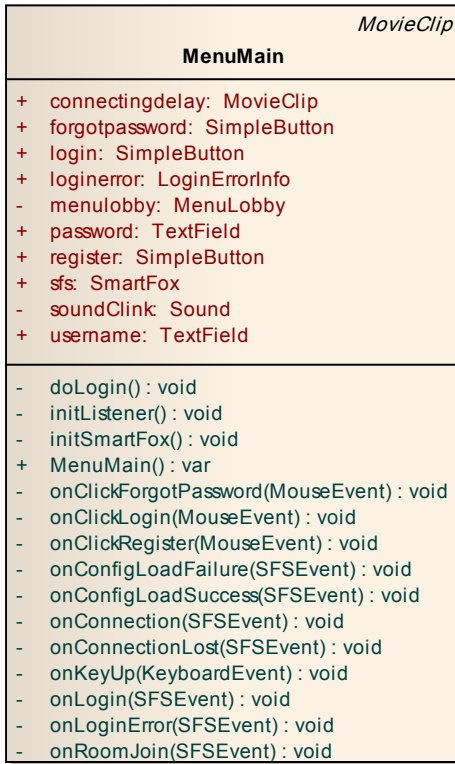
DAFTAR PUSTAKA

- [1] U. & R. Choironi. (2011, Apr.) Petualangan Berburu Bintang Hijaiyah. [Online]. <http://www.gameedukasi.com/2011/01/petualangan-berburu-bintang-hijaiyah/>
- [2] N. Lovell. (2009, Sep.) The six secrets of Farmville's success. [Online]. <http://www.gamesbrief.com/2009/09/six-secrets-of-farmvilles-success-and-33-million-people-agree/>
- [3] A. Mahdan, *Rancang Bangun Permainan Waktu Nyata dengan Banyak Pemain Berbasis Web "Solus Saga"*. Surabaya, Indonesia: Institut Teknologi Sepuluh Nopember, 2011.
- [4] A. R. Jannah, *Game Edukasi Bahasa Arab untuk Anak Berbasis Desktop*. Surabaya, Indonesia: Politeknik Elektronika Negeri Surabaya, 2011.
- [5] J. Juul. (2003, Nov.) The Game, the Player, the World: Looking for a Heart of Gameness. [Online]. <http://www.jesperjuul.net/text/gameplayerworld/#sdfootnote1anc>
- [6] gotoAndPlay(). (2010) SmartFoxServer Showcase. [Online]. <http://www.smartfoxserver.com/showcase/>
- [7] G. Grossman and H. Emmy. (2006, Jun.) ActionScript 3.0 Overview. [Online]. http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html
- [8] gotoAndPlay(). (2010) SmartFoxServer Overview. [Online]. <http://www.smartfoxserver.com/overview.php>

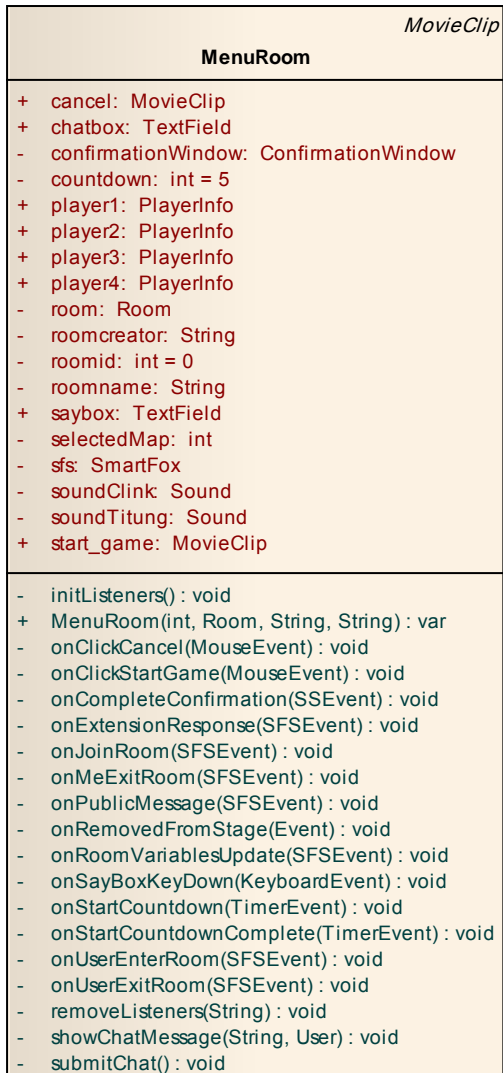
[Halaman ini sengaja dikosongkan]

LAMPIRAN A

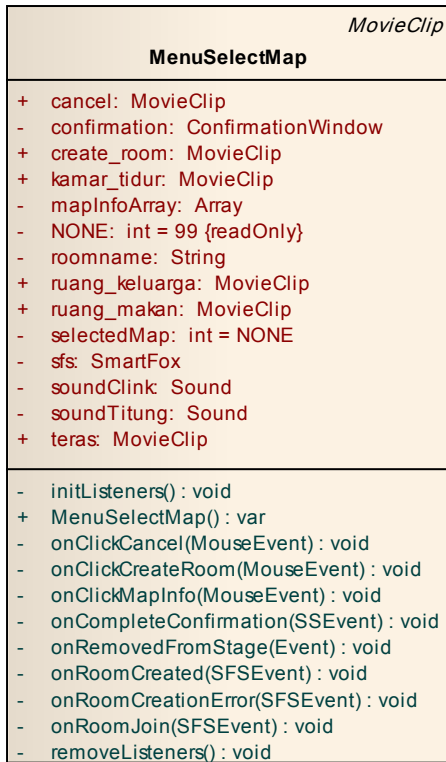
DIAGRAM KELAS PERMAINAN “YUK KITA CARI”



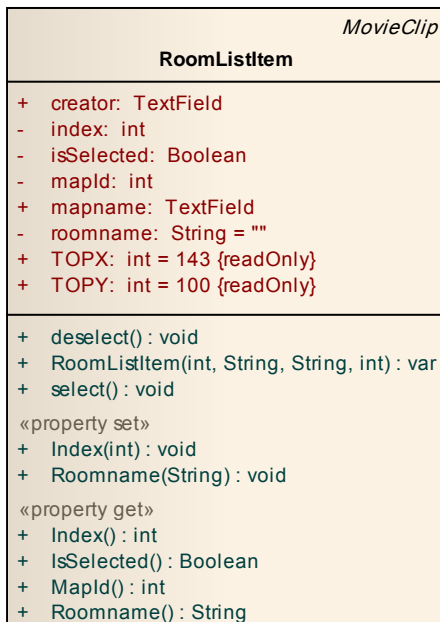
Gambar A.1 Diagram Kelas MenuMain



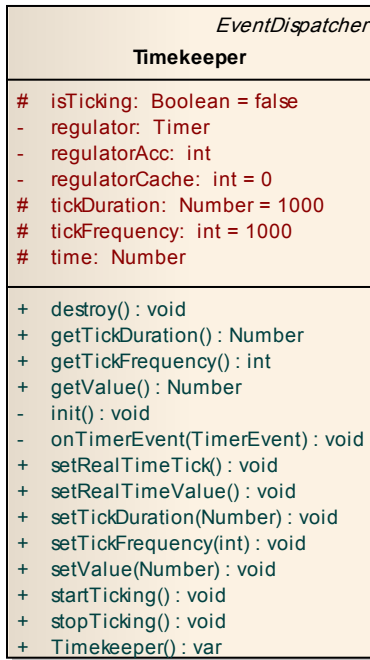
Gambar A.2 Diagram Kelas MenuRoom



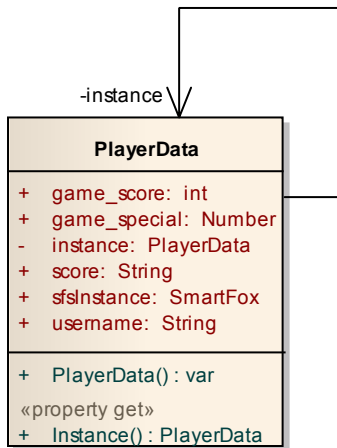
Gambar A.3 Diagram Kelas MenuSelectMap



Gambar A.4 Diagram Kelas RoomListItem



Gambar A.5 Diagram Kelas TimeKeeper



Gambar A.6 Diagram Kelas PlayerData

LAMPIRAN B

KODE SUMBER REGISTER.PHP

Kode Sumber B.1 register.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<?
//require_once 'MailSender.php';
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Register</title>
<?php
    // put your code here
    $iscaptcha=false;
    $ispost=false;
    $istrue=false;
    $isavailable=false;

    if(isset ($_POST["Register"]))
    {
        $ispost=true;
        session_start();

if(md5($_POST['pin'])==$_SESSION['image_random_value'])
    {
        //=====CAPTCHA SUCCESS
        $iscaptcha=TRUE;

        //=====Check Username
        $con=mysql_connect("localhost", "root",
"root");

        if (!$con)
        {
            die ('Could not connect: '.
mysql_error());
        }

        mysql_select_db("database_yukari", $con);

        $result=mysql_query("select
count(a.username) as c from (select username from `user`
where username='".$_POST['username']."' ) as a");
        while($row= mysql_fetch_array($result))
        {
```

```

        if($row['c']==0)
        {
            $isavailable=TRUE;
        }
        else
        {
            $isavailable=FALSE;
        }
    }

    mysql_close($con);

    //=====Generate Key
    $key=md5($_POST['username']);

    //=====Final Check
    if($iscaptcha==TRUE && $ispost==TRUE &&
$isavailable==TRUE)
    {
        $istrue=TRUE;
    }

    //=====Database Insert
    if($istrue)
    {
        $con=mysql_connect("localhost", "root",
"root");
        if(!$con)
        {
            echo "error pek";
            die('Could not connect: '.
mysql_error());
        }

        mysql_select_db("database_yukari",$con);

        mysql_query("INSERT INTO
`database_yukari`.`user` (`username`, `password`, `email`,
`secretquestion`, `secretanswer`) VALUES
('".$_POST['username']."' ,
'".$_md5($_POST['password'])."', '".$_POST['email']."' , '".$_POS
T['secretquestion']."' , '".$_md5($_POST['secretanswer'])."');"
;
        //
        echo "proses insert lewat";

        mysql_close($con);
    }
}

```

```

    }

    ?>

<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>

<div class="container">

<div id="header">

<div id="contact details">
<p><a href="#">Mohammad F K Umam</a></p>

</div><!--end contact details-->

</div><!-- end header -->
<div style="clear: both"></div>

<div id="main">

<div id="content">

<div class="text">
<h2>Register for Yukari Multiplayer</h2>
</div><!--end text-->
<div class="text">
<?if(!$istrue)
                                {?>
<form name="register" method="post" action="register.php"
onsubmit="return validateForm()">

<h3>Username (4-8 character alphabet only)<?if(!$isavailable
&& $ispost)
                                {?>

<h4>Username is not available</h4>

<?

$con=mysql_connect("localhost","root","root");
                                if(!$con)
                                {
                                        die('Could not connect:
'. mysql_error());
                                }

```



```

<h4>False captcha!</h4><?}?>
</h3>
<h3></h3>

<h3><input type="text" size="20" name="pin"/></h3>
<h3><input type="submit" name="Register" value="Register"
/></h3>
</form><!--END OF FORM-->

<?
                                }
                                else if($istrue)
                                {
                                ?>
<h3>Your registration is complete, now you can use your
account, <?=$_POST['username']?> !</h3>

<?}?>
</div>

</div>

</div>

</div>

</body>
</html>

```

[Halaman ini sengaja dikosongkan]

LAMPIRAN C

KODE SUMBER FORGOTPASSWORD.PHP

Kode Sumber C.1 forgotpassword.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<?
require_once 'MailSender.php';
?>

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Forgot Password</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>

<body>
<?php

    $iscaptcha = false;
    $ispost = false;
    $isexist = false;
    $istrue = false;
    $newpass = "";

    if(isset ($_POST["forgotpassword"]))
    {
        $ispost = true;
        session_start();
        if(md5($_POST['pin']) ==
$_SESSION['image_random_value'])
        {
            //=====captcha success
            $iscaptcha = true;
            //=====check username
            $con =
mysql_connect("localhost","root","root");
            if(!$con)
            {
                die('Could not connect:'.
mysql_error());
            }

            mysql_select_db("database_yukari", $con);
```

```

        $result = mysql_query("select
count(a.username) as c from (select username from `user`
where username='".$$_POST['username']."' and
email='".$$_POST['email']."' ) as a");
        while($row = mysql_fetch_array($result))
        {
            if($row['c']==1) $isexist = true;
            else $isexist = false;
        }

        mysql_close($con);

        if($iscaptcha==true && $ispost==true &&
$isexist==true)
        {
            $con =
mysql_connect("localhost","root","root");
            if(!$con)
            {
                die('Could not connect:'.
mysql_error());
            }

            mysql_select_db("database_yukari",$con);

            $result = mysql_query("select email as
em, secretquestion as sq, secretanswer as md from `user`
where username='".$$_POST['username']."'");
            while($row =
mysql_fetch_array($result))
            {
                if($row['sq']==$_POST['secretquestion'] && $row['md'] ==
md5($_POST['secretanswer']) && $row['em'] ==
$_POST['email'])
                {
                    $istrue = true;
                }
                else
                {
                    $istrue = false;
                }
            }

            mysql_close($con);
        }
        if($istrue==true)
        {

```

```

        $newpass =
$newpass.rand(1000,2000).rand(1000,2000);
        $con =
mysql_connect("localhost","root","root");
        if (!$con)
        {
            die('Could not connect:
'.mysql_error());
        }

mysql_select_db("database_yukari",$con);
        mysql_query("UPDATE `user` set
`password`='".md5($newpass)."' where
username='".$_POST['username']."'");
        mysql_close($con);
    }
}

?>
<div class="container">
<div id="header">
<div id="logo">
<a href="index.html"></a>
</div>

<div id="contact_details">
<p><a href="#">Mohammad F K Umam</a></p>
<p><a href="#">Teknik Informatika</a></p>
<p><a href="#">Institut Teknologi Sepuluh Nopember</a></p>
</div><!--end contact details-->

</div><!--end header-->
<div style="clear: both"></div>

<div id="main">
<div id="content">
<div class="text">
<h2>Lupa Kata Kunci Permainan Yuk Kita Cari</h2>
</div><!--end text-->
<div class="text">
<?if($istrue==false)
        {?>
<?if($isexist==false && $ispost)
        {?>
<h4>data tidak tersedia, masukkan kembali data anda</h4><?
        }?>

```

```

<form name="Forgotpassword" method="post"
action="forgotpassword.php">
<h3>Username</h3>
<h3><input type="text" name="username" size="20"
maxlength="8"/></h3>
        <h3>E-mail </h3>
        <h3><input type="text" name="email"
size="20" maxlength="20"/></h3>
        <h3>Secret Question </h3>
        <h3>
        <select name="secretquestion">
                <option selected="selected"
value="What is the name of your best friend from
childhood?">What is the name of your best friend from
childhood?</option>
                <option value="What was the
name of your first teacher?">What was the name of your
first teacher?</option>
                <option value="What is the
name of your manager at your first job?">What is the name
of your manager at your first job?</option>
                <option value="What was your
first phone number?">What was your first phone
number?</option>
                <option value="What is your
vehicle registration number?">What is your vehicle
registration number?</option>
        </select>
        </h3>
        <h3>Secret Answer </h3>
        <h3><input type="text" name="secretanswer"
size="30" maxlength="100"/></h3>
        <h3>Input text below:
<?if($iscaptcha==false && $ispost){?><h4>False
captcha!</h4><?}></h3>
        <h3></h3>
        <h3><input type="text" size="20"
name="pin" /></h3>
        <h3><input type="submit"
name="forgotpassword" value="Request New Password" /></h3>
</form>
<?
        }
        else if($istrue==true)
        {?>
<h3>your password has been reset, this is your new
password, please change your password quickly!
<?echo($newpass)?> !</h3>

```

```
<?
                                }?>
</div><!--end text-->
</div><!--end content-->
</div><!--end main-->
</div><!--end class container-->
</body>
</html>
```

[Halaman ini sengaja dikosongkan]

LAMPIRAN D

KODE SUMBER PENGECEKAN

Kode Sumber D.1 Pengecekan Registrasi Tahap Pertama

```
<script type="text/javascript">
function validateForm()
{
    var usr=document.forms["register"]["username"].value
    if (usr==null || usr.length<4 || usr.length>8)
    {
        alert("Username must be 4-8 character alphabet only");
        return false;
    }
    var re = /^[a-zA-Z]$/;
    if(!(re.test(usr))){
        alert("Username must be 4-8 character alphabet
only");
        return false;
    }

    var pass=document.forms["register"]["password"].value
    if (pass==null || pass.length<6 || pass.length>20)
    {
        alert("Password must be 6-20 character");
        return false;
    }

    var
    repass=document.forms["register"]["repeatpassword"].value
    if (repass!=pass)
    {
        alert("Password does not match");
        return false;
    }

    var x=document.forms["register"]["email"].value
    var atpos=x.indexOf("@");
    var dotpos=x.lastIndexOf(".");
    if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
    {
        alert("Not a valid e-mail address");
        return false;
    }

    var ans=document.forms["register"]["secretanswer"].value
    if (ans==null)
    {
        alert("Please fill the secret question and answer");
    }
}
```

```

        return false;
    }
}
</script>

```

Kode Sumber D.2 Pengecekan Registrasi Tahap Kedua

```

<?php
    // put your code here
    $iscaptcha=false;
    $ispost=false;
    $istrue=false;
    $isavailable=false;

    if(isset ($_POST["Register"]))
    {
        $ispost=true;
        session_start();

if(md5($_POST['pin'])==$_SESSION['image_random_value'])
    {
        //=====CAPTCHA SUCCESS
        $iscaptcha=TRUE;

        //=====Check Username
        $con=mysql_connect("localhost", "root",
"root");

        if(!$con)
        {
            die ('Could not connect: '.
mysql_error());
        }

        mysql_select_db("database_yukari", $con);

        $result=mysql_query("select
count(a.username) as c from (select username from `user`
where username='".$_POST['username']."' ) as a");
        while($row= mysql_fetch_array($result))
        {
            if($row['c']==0)
            {
                $isavailable=TRUE;
            }
            else
            {
                $isavailable=FALSE;
            }
        }
    }
}

```



```

mysql_close($con);

//=====Generate Key
$key=md5($_POST['username']);

//=====Final Check
if($iscaptcha==TRUE && $ispost==TRUE &&
$isavailable==TRUE)
{
    $istrue=TRUE;
}

//=====Database Insert
if($istrue)
{
    $con=mysql_connect("localhost", "root",
"root");

    if(!$con)
    {
        echo "error pek";
        die('Could not connect: '.
mysql_error());
    }

    mysql_select_db("database_yukari",$con);

    mysql_query("INSERT INTO
`database_yukari`.`user` (`username`, `password`, `email`,
`secretquestion`, `secretanswer`) VALUES
('".$_POST['username']."',
'".$_md5($_POST['password'])."', '".$_POST['email']."', '".$_POS
T['secretquestion']."', '".$_md5($_POST['secretanswer']).'');"
;
//        echo "proses insert lewat";

    mysql_close($con);
}

}

?>

```

[Halaman ini sengaja dikosongkan]

LAMPIRAN E

KODE SUMBER APLIKASI KLIEN

Kode Sumber E.1 Proses Cek Pemenang pada Kelas PlayGame

```
private function checkWinner():void
{
    //gamechat.chatbox.appendText("Complete = " +
    timerinfo.getBoolTimerComplete().toString() + " ");
    if (bufferMatched.length <= 0 || isForceTimesUp == true
    || timercountdown.getIsTimesUp() == true)
    {
        isPause = true;

        //di isi lagi array nya biar g kosong
        bufferMatched = bgwithunit.unitArray;

        //request cek pemenang
        var param:ISFSObject = new SFSObject();
        param.putInt("msg", Const.GAME_REQUEST_WIN);
        param.putUtfString("username",
        PlayerData.Instance.username);
        param.putInt("score", PlayerData.Instance.game_score);
        var request:ExtensionRequest = new
        ExtensionRequest(Const.GAME_MESSAGE, param);
        sfs.send(request);
        winDialogOnCreator = "Alhamdulillah, semua komplit";
        winDialogOnPlayer = "Alhamdulillah, tetap semangat";
        gamechat.chatbox.appendText("TEST CHECK");
    }
}
```

Kode Sumber E.2 Kode Untuk Menangani Respon Server

```
//response thd server (komunikasi)
private function onExtensionResponse(e:SFSEvent):void
{
    var params:ISFSObject = e.params.params as ISFSObject;
    switch(params.getInt("msg"))
    {
        case Const.GAME_USER_REPORT:
            //pengecekan, jika ad yg g beres
            //brarti d list lg ndak ad user
            if (usercount >= userlist.length)
            {
                break;
            }
            if (userlist.indexOf(params.getUtfString("username"))
```

```

== -1)
    {
        break;
    }
    usercount++;
    if (usercount == userlist.length)
    {
        startTheGame();
    }
    break;
case Const.SERVER_SHUTDOWN:
    isPause = true;
    confirmationWindow = new
ConfirmationWindow(ConfirmationWindow.CONFIRM, "Kamu
terputus dari jaringan, alasan: server mati");
    confirmationWindow.addEventListener(SSEvent.COMPLETE,
onCompleteConfirmationWindowAfterServerShutdown);
    addChild(confirmationWindow);
    break;

    //check unit yg telah d cocokkan & lakukan perubahan
thd unit tsb
    case Const.GAME_MATCHED_UNIT:
        var matchedIndex:int = params.getInt("matched");
        var stringName:String =
params.getUtfString("username");
        //tampilan log box
        gamelog.logbox.htmlText += "<font
color='0x000000'><b>" + "Pemain " + stringName + " " +
"mencocokkan benda " +
bgwithunit.unitArray[matchedIndex].currentFrameLabel + " " +
"</b></font>";
        gamelog.logbox.scrollV = gamelog.logbox.numLines - 2;
        objectChanger(stringName, matchedIndex);
        break;

    case Const.GAME_REQUEST_WIN:
        //pilih skor yg paling besar
        if (bufferScore <= PlayerData.Instance.game_score)
        {
            bufferScore = PlayerData.Instance.game_score;
            bufferName = PlayerData.Instance.username;
        }
        if (bufferScore <= params.getInt("score"))
        {
            bufferScore = params.getInt("score");
            bufferName = params.getUtfString("username");
        }

```

```

        //pantulkan ke ekstensi agar d kirim ke semua pemain
        var putparamwin:ISFSObject = new SFSObject();
        putparamwin.putInt("msg", Const.GAME_WIN);
        putparamwin.putUtfString("username", _bufferName);
        putparamwin.putInt("score", _bufferScore);
        var putreqwin:ExtensionRequest = new
ExtensionRequest(Const.GAME_MESSAGE, putparamwin);
        sfs.send(putreqwin);
        gamechat.chatbox.appendText(" GAME_REQUEST_WIN ");
        isForceTimesUp = true;
        break;

        case Const.GAME_WIN:
            //tangkap pantulan dr ekstensi
            gamechat.chatbox.appendText(" GAME_WIN ");
            var winnerName:String =
params.getUtfString("username");
            var winnerScore:int = params.getInt("score");
            //tampilkan dialog
            if (isCreator)
            {
                var tx:Timer = new Timer(1000, 1);
                tx.start();
            }
            showWinDialog(winDialogOnCreator, winnerName,
winnerScore);
            gamechat.chatbox.appendText(" SHOW_DIALOG ");
            _bufferScore = 0;
            _bufferName = null;
            break;

            case Const.GAME_BROADCAST_WIN:
                gamechat.chatbox.appendText( "GAME BROADCAST WIN" );
                var winNamePlayer:String =
params.getUtfString("winner");
                var winScorePlayer:int = params.getInt("winscore");
                isForceTimesUp = true;
                if (isCreator)
                {
                    //do nothing
                    gamechat.chatbox.appendText( " BROADCAST " );
                }
                if(!isCreator)
                {
                    showWinDialogPlayer(winDialogOnPlayer,
winNamePlayer, winScorePlayer);
                }

                break;

```

```
        case Const.GAME_REQUEST_TIMER:
            break;
    }
}
```

BIODATA PENULIS



Penulis lahir di Malang pada bulan Mei tahun 1988. Penulis menempuh pendidikan dasar mulai kelas 1 hingga kelas 6 di MIN Malang I. Untuk pendidikan menengah penulis tempuh di SMP Negeri 3 Malang. Kemudian penulis melanjutkan pendidikan di SMA Negeri 3 Malang. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember Surabaya.

Selama kuliah penulis aktif di berbagai organisasi mahasiswa mulai dari Studi Islam Teknik Komputer, LSM LP3M, Badan Koordinasi Lembaga Da'wah Kampus Jawa Timur, juga Gerakan Mahasiswa Pembebasan Jawa Timur. Saat ini penulis tercatat sebagai aktivis Hizbut Tahrir Chapter ITS.

Penulis dalam menyelesaikan pendidikan S1 mengambil bidang minat Rekayasa Perangkat Lunak (*Software Engineering*) dan memiliki ketertarikan di bidang *Software Development*. Penulis dapat dihubungi melalui email: umamlearn@gmail.com.

[Halaman ini sengaja dikosongkan]